

Mediatability: Estimating User Effort to Mediate Between Two XML Schemas

Karthik Gomadam¹, Ajith Ranabahu¹, Kunal Verma²,
Lakshmith Ramaswamy³ and Amit P. Sheth¹
kgomadam@gmail.com, {ranabahu.2, amit.sheth} @wright.edu,
k.verma@accenture.com, laks@cs.uga.edu

¹Knoesis Center, Dayton, OH, USA. ²Accenture Technology Labs, CA USA.

³University of Georgia, GA USA.

Abstract

Mediation and integration of data is one of the important challenges because of the ever-increasing number of services on the Web and heterogeneities in their data representations. Towards addressing this challenge, we introduce a new measure called mediatability. Mediatability is a quantifiable and computable metric that measures the degree and complexity of human involvement in XML schema mediation. We also present an efficient algorithm to compute mediatability. We provide an experimental study to analyze the impact of having semantic annotations in determining the ease of mediation between two schemas. We validate our approach by comparing the mediatability scores generated by our system against that of user perceived difficulty in mediation. We also evaluate the scalability of our system in both client and server contexts.

Web

1 Introduction

The increased adoption of the REpresentational State Transfer paradigm [5] has made it easier to create and share services on the Web. RESTful services often take the form of RSS/Atom feeds and AJAX based light weight services. The XML based messaging paradigm of RESTful services has made it possible to bring discrete data from services together and create more meaningful data sets. This is being referred to as building a mashup. A mashup is the creation of a new Web application using two or more existing Web application interfaces. Some of the problems that are viewed as an impediment for developers to create mashups are : 1) the programming skill required to develop such applications (largely due to complexity of languages such as javascript) and 2) the arduous task of mapping the output of one service to the input of another. Frameworks such

as Google Mashup Editor¹ and IBM Sharable Code² have addressed the first problem with reasonable success by creating programming level abstractions. However, little work has been towards helping the developers in the task of data mediation.

The importance of understanding and addressing the problem of data mediation in distributed systems is underscored by the volume of research in matching and mapping heterogeneous data. *Matching* is the task of finding correspondences between elements in schemas or instances. Once the corresponding elements are identified, *mapping* defines the rules to transform elements from one schema into another. Matching and mapping have been well studied by various researchers including [7], [16] and [8] in different contexts. Considerable research effort has gone into creating frameworks that attempt automated and semi-automated matching and mapping of heterogeneous data. Much of these efforts however, have yielded limited success and developers are often left with the hard task of performing the mediation manually.

The end goal of traditional schema matching has been to establish semantic similarity between schema elements. However, semantic equivalence does not guarantee inter-operation. Depending on the heterogeneities between the schemas, mediation is harder or even impossible to automate [16]. Even when mediation is manual, it is hard to estimate the complexity for a developer to perform mediation between the two schemas. The goal of this paper is go a step beyond matching and define mediatability as a measure of the degree and complexity of human involvement. We believe that such a measure would help users in selecting services, especially in the REST services scenario, where often one has to choose from a plethora of services that offer the same features with very little separation.

Our experience with IBM Sharable Code [9] largely

¹<http://editor.googlemashups.com/editor>

²<http://services.alphaworks.ibm.com/iscscore>

motivated this work in quantifying ease of mediation. In creating the data components for the IBM sharable code mashups, a significant amount of effort was needed to pick the correct data elements, often from large and complex schemas. This is illustrated by the fact that the popular REST API directory, programmableWeb.com³, returns 71 services for the search keyword *mapping*. Most real world services (for example Amazon⁴, Microsoft Live⁵) model a rich schema, making them large and verbose. We believe based on our experience on creating real world mashups [19], having a quantifiable measure of the degree and complexity of human involvement in mediation, would serve as a useful metric in the selection of services.

The paper makes two unique contributions.

- First, we introduce the concept of mediatability as an indicator of the degree and complexity of human involvement in mediation between two schemas. Further, we provide a quantifiable definition of mediatability that takes into account the element level similarity and the structural similarity of the two XML schemas.
- Second, we provide an efficient two pass algorithm for computing the mediatability. The similarities are computed in the top-down pass and the mediatability is computed in the bottom-up pass. Further, we discuss an optimization technique to get a better average case time complexity.

There has been activity in semantically annotating schemas and since it is a high indicator of semantic similarity between 2 elements, it is valuable to see what this brings to the problem of computing mediatability. We provide an experimental study to analyze the impact of having semantic annotations in determining the ease of mediation between two schemas. We validate our approach by comparing the mediatability scores generated by our system against that of user perceived difficulty in mediation. We also evaluate the scalability of our system.

2 Motivation

We present an example that illustrates the need for and the use of mediatability. In this example, we consider a developer trying to create a mashup in which one of the services is an image search service. Examples of such mashups can be found at [22]. Services such as Microsoft live search and Yahoo image search return image results for a given search string and the developer has to choose one of services. Snippets of the Yahoo image search and Microsoft live search result schemas along with the desired target schema of the developer is illustrated in Figure 1. For the

³<http://www.programmableWeb.com/apitag/?q=mapping, 03/14/2008>

⁴<http://soap.amazon.com/schemas2/AmazonWebServices.wsdl>

⁵<http://soap.search.msn.com/Webservices.asmx?wsdl>

purposes of the example, we consider the schemas of Live and Yahoo image search to be the source schemas. As we can see from Figure 1, the live result schema is nested and deep, while the Yahoo schema is shallow. Given that both Live and Yahoo image search services return a set of images for a given search query, one of the metrics that can help in differentiating between the two services is the ease with which the developer can mediate between the schema of the service provider and the target schema. Mediatability is the measure of the ease of performing this mediation.

In the next section, we define mediatability and illustrate with an example based on the source and target schemas illustrated in Figure 1.

3 Mediatability: Definition and Computation

In this section we present the conceptual definition of mediatability between two schemas and discuss our approach to calculating a concrete quantifiable metric. Mediatability is defined as the measure of the degree and complexity of human involvement in mediation between two schemas based on their semantic and structural similarities. The value of mediatability between two schemas lies between 0 (hardest to mediate; indicates lot of human effort) and 1 (easy to mediate; indicates little effort). Formally, mediatability between a target schema T and a source schema S is defined as

$$\sigma(T, S) = x : x \in [0, 1]$$

While we believe that such a notion can be defined between any two schemas (databases, ontologies), in this paper we focus on computing the mediatability for XML schemas. The conceptual definition of mediatability cannot be used directly. We present a computable and quantifiable definition of mediatability between two schemas and discuss our approach toward calculating mediatability between two schemas.

3.1 Overview

Mediatability between two schemas is computed by first computing the the mediation similarity between the two elements of the two schemas. The mediation similarity between two elements is a function of their element similarity and structural similarity. Element similarity between two elements is a function of *Semantic Similarity*, *Wordnet Similarity*, *Lexical Similarity* and *Type Similarity*.

To compute the structural similarity, we first identify the nearest similar ancestor of the two elements. The nearest similar ancestor between an element e_i^t in the target schema and an element e_j^s in the source schema is a pair of elements e_p^t in target schema and e_q^s in source schema such that e_q^s belongs to the similarity set of e_p^t and e_p^t is the nearest such

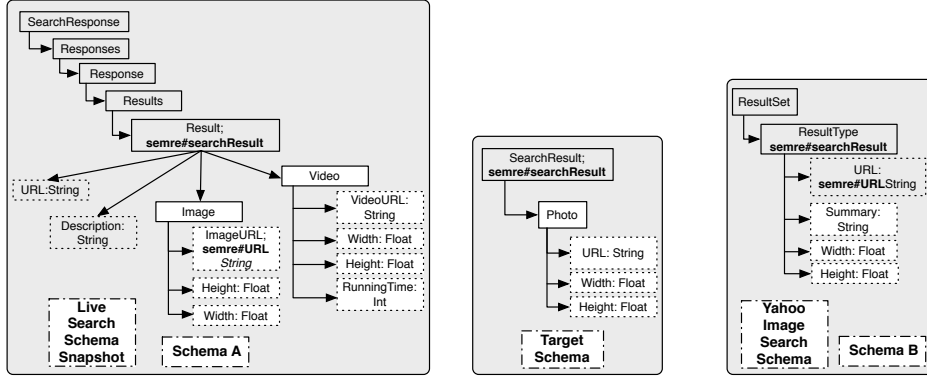


Figure 1. Search Services and Search Request Schemas

element to e_i^t in the target schema. The mediation similarity between e_i^t and e_j^s is defined as a measure of the structural and the semantic similarity between the two elements and is a function of the element similarity between them, the mediation similarity between their nearest similar ancestor elements and the distance between the elements and their NSA.

The mediatability between an element in the target schema and an element in the source schema is computed in a recursive manner by computing the mediatability between the elements in the two schemas and is computed in a bottom-up manner, beginning with the leaf elements and terminating at the root element. This is illustrated in Figure 3 (b). The mediatability between two elements as the average mediatability between their respective child elements. If an element in the target schema is a leaf node, then the mediatability between that element and an element in the source schema is same as the mediation similarity between them. The formal definition and a detailed discussion about computing the mediatability is presented in section 3.5.

We now present our approach for computing the mediatability in detail.

3.2 Computing Element Similarity

Converting the source and target schemas into schema hierarchy trees is the first step in computing the mediatability. The schema hierarchy trees are created by converting each element in the XML schema to a node that contains the name of the XML element, the semantic annotation on that element and the XML data type of the element. If the type of an XML element is a complex type, then the data type property of that node is empty. Complex types and references are expanded in place. The in place expansion allows us to model the schema as a tree and removes the links between different elements in the schema. In our discussion we denote the source schema hierarchy tree as H_s and the target schema hierarchy tree as H_t . Elements in the source schema hierarchy tree are denoted by e_j^s and the elements

in the target schema hierarchy tree are denoted by e_i^t .

Once the schema hierarchy trees are constructed, we compute the element similarity between the elements in H_t and H_s . This is computed in a top-down manner starting with the root of the target schema hierarchy. To compute the element similarity, we compare the elements in the target and source trees. The element similarity computation is illustrated in Figure 2(a).

- *Semantic Similarity:* If semantic annotations are present in both the target and source elements, concept similarity is calculated by computing the relationship between the concepts in the semantic model referenced by the annotations. If the relationship between the concepts is one of subclass, superclass or equivalence, then the semantic relationship is used in defining the semantic similarity. Since the SearchResult element the target schema and the Result element of schema A in Figure 2(a) have annotations and the annotations are equivalent, the semantic similarity between them is 1. This is defined as,

$$S_{sim}(e_i^t, e_j^s) = \begin{cases} W_{sub} & t_a \sqsubseteq t_s \\ 1 & t_a \equiv t_s \\ W_{sup} & t_a \supseteq t_s \\ 0 & t_a, \text{ other relationships} \end{cases}$$

where W_{sub} and W_{sup} are scores assigned to subclass and superclass relationships and t_a and t_s are the ontology concepts referenced by the source and target annotations respectively.

- *Wordnet Similarity:* If the semantic similarity cannot be computed or is zero, we compute the wordnet similarity between the element names based on the relationship between them in Wordnet [13]. In Figure 2(a), the Photo element of the target schema and the Image element in schema A are not annotated. Hence the similarity between them is computed using wordnet. Since

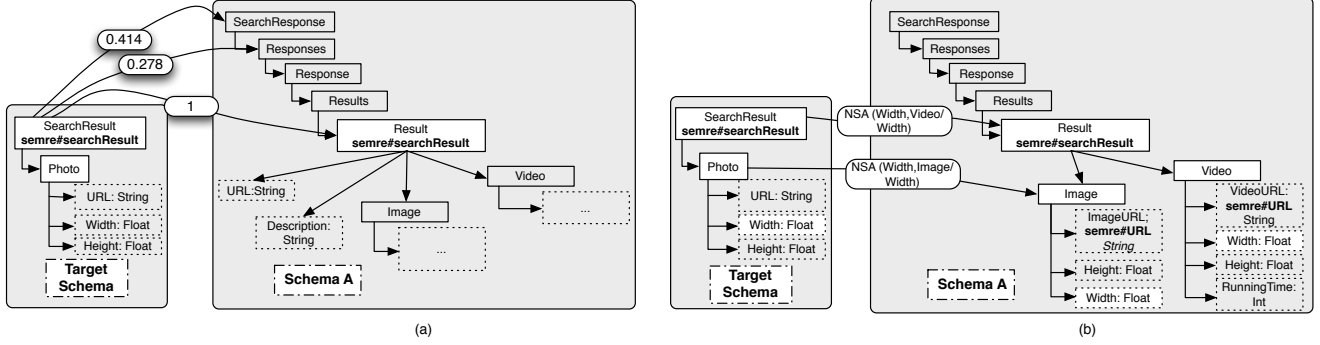


Figure 2. (a) Computing Element Similarity (b) Nearest Similar Ancestor

they are synonyms, their wordnet similarity is 1. The wordnet similarity is defined as,

$$W_{sim}(e_i^t, e_j^s) = \begin{cases} \frac{W_{hyppo}}{d} & e_i^t \text{ hyponym of } e_j^s \\ 1 & e_i^t \text{ synonym of } e_j^s \\ \frac{W_{hyype}}{d} & e_i^t \text{ hypernym of } e_j^s \\ 0 & \text{, otherwise.} \end{cases}$$

where W_{hyppo} and W_{hyype} are scores assigned to hyponym and hypernym relationships respectively and d is the depth of the relationship.

- **Lexical Similarity:** If both the semantic similarity and the wordnet similarity is zero, we compute the lexical similarity between the element names using edit distance. This is denoted by L_{sim} . In the example illustrated in Figure 2(a), the lexical similarity between the SearchResult element of the target schema and the SearchResponse element of Schema A is computed, since their semantic and wordnet similarities are zero.
- **Type Similarity:** The type similarity ($T_s(e_i^t, e_j^s)$) between the elements is calculated by comparing the xsd:type of the elements and the similarity value is based on the two types being compared. If the types match, then the type similarity is exact.

We define the element similarity as,

$$E_s(e_i^t, e_j^s) = C_s(e_i^t, e_j^s)T_s(e_i^t, e_j^s) \quad (1)$$

$$\text{where, } C_s(e_i^t, e_j^s) = \begin{cases} S_{sim} & \text{, if semantic similarity} \\ W_{sim} & \text{, if wordnet similarity} \\ L_{sim} & \text{, if lexical similarity} \end{cases}$$

3.3 Factoring Structural Similarity

In computing the element similarity, we only consider the similarity between the semantic annotations and the element names along with the type similarity. The structural

similarity, which plays a crucial role in determining the element similarity cannot be ignored. For example, the width element, which is a child of element of the photo element in the target schema in Figure 2(b) would match completely with the width elements contained in both Image and Video elements in schema A, if one were to only consider the annotation and type similarities. However, when we consider the ancestry of both the elements, we find that the width element under the video element and the width element in the target schema do not have a similar ancestor until the SearchResult and Result elements respectively. However, the parent elements of the width element under the image element and the width element in the target schema are similar. Factoring this information, we can say that the width element under the image element is more similar to the one in the target schema. To factor in the ancestry, we define the nearest similar ancestor between an element in the target hierarchy and an element in the source hierarchy.

The *Nearest Similar Ancestor* ($NSA(e_i^t, e_j^s)$) is the pair of elements (e_p^t, e_q^s) such that e_q^s belongs to the similarity set of e_p^t . This is defined as,

$$NSA(e_i^t, e_j^s) = (e_p^t, e_q^s) : e_q^s \in S_{e_p^t} \wedge e_q^s \text{ is the nearest ancestor of } e_i^t. \quad (2)$$

where $S_{e_p^t}$ is the similarity set of e_p^t . The similarity set of an element is defined later in the section. The definition of nearest similar ancestor between two elements in a hierarchy is inspired by the definition of nearest common ancestor proposed by Dov and Tarjan in [6].

3.4 Computing mediation similarity

Using the element similarity and the nearest similar ancestor, we define the mediation similarity between e_i^t and e_j^s . Two elements may have an element similarity of 1, but if there is very little structural similarity between the two schemas, the mediation similarity would be significantly lower. The structural similarity depends on the level of the target and source elements in the respective hierarchy trees

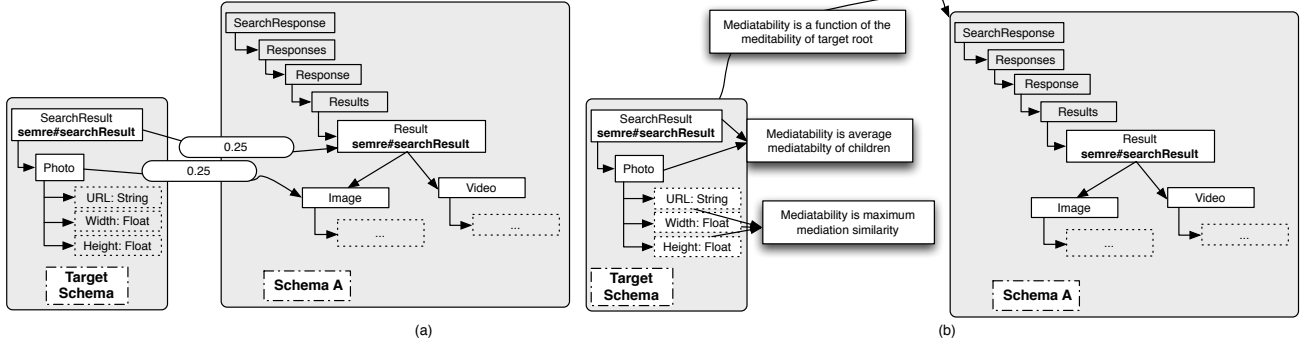


Figure 3. (a) Computing mediation similarity (b) Mediatability Computation

from their nearest similar ancestors. If the NSA (e_i^t, e_j^s) exists, the mediation similarity is measured by factoring their element similarities, the mediation similarity between the NSA elements and the distance between e_i^t, e_j^s and their respective ancestors in the NSA. If there is no similar ancestor between e_i^t and e_j^s , the mediation similarity is computed factoring in the element similarity and the depth of the elements in the hierarchy. If either of the two elements is the root element, then its depth is taken to be 1. The formulae for computing the mediation similarity is.

$$OS(e_i^t, e_j^s) = \begin{cases} E_s(e_i^t, e_j^s) \frac{OS(e_p^t, e_q^s)}{d_{ip}d_{jq}} \\ E_s(e_i^t, e_j^s) \frac{1}{d_i d_j}, \text{ if NSA is empty.} \end{cases} \quad (3)$$

where d_{ip} is the depth of e_i^t from its nearest similar ancestor, d_{jq} is the depth of the e_j^s from its nearest similar ancestor, d_i is the depth of e_i^t and d_j is the depth of e_j^s . We now illustrate with an example. Consider the target schema and schema in Figure 3 (a). The element similarity between the SearchResult element in the target schema and the Result element in schema A is 1. Now the depth of the Result element in schema A is 4, while the SearchResult element in the target schema is the root and hence its depth is taken to be 1. The mediation similarity between the two elements is 0.25. Now we consider the Photo element of the target schema and the Image element of schema A. The NSA(Photo, Image) = (SearchResult, Result). The element similarity between the photo and the image elements is 1 and the mediation similarity of the NSA elements is 0.25, from the above. Using the formula for mediation similarity defined in equation 3, the mediation similarity between photo and the image element is 0.25.

The similarity set of e_i^t ($S(e_i^t)$) is the set of elements e_j^s in the source schema that have the maximum similarity value with e_i^t .

$$S(e_i^t) = \{e_j^s : OS(e_i^t, e_j^s) \text{ is maximum}\} \quad (4)$$

As an example, the similarity set of the photo element of the target schema is { Image }.

The mediation similarity coefficient of a target element e_i^t is the maximum mediation similarity value between e_i^t and any source element.

$$OS_C(e_i^t) = \begin{matrix} \text{maximum mediation similarity value between} \\ e_i^t \text{ any source element.} \end{matrix} \quad (5)$$

As an example, in Figure 3(a), the mediation similarity coefficient of the Photo element of the target schema is 0.25.

3.5 Calculating Mediatability

We now discuss the calculation of the mediatability between two schemas. While element similarity is computed in a top-down manner, mediatability is computed in a bottom up manner, beginning with the leaf elements of the target schema.

Mediatability of an element e_i^t in the target schema is denoted by σ . If an element e_i^t is a leaf element, the mediatability of e_i^t its mediation similarity coefficient defined in equation 5.

$$\sigma(e_i^t) = OS_C(e_i^t) \quad (6)$$

The width element in the target schema in Figure 3(b) is a leaf element. Hence its mediatability is same as its mediation similarity coefficient, which is 0.25.

For each e_i^t that is not a leaf element, the mediatability of e_i^t defined as the average of mediatability between its immediate children.

$$\sigma(e_i^t) = \frac{1}{z} \sum_{m=0}^z \sigma(e_m^t) \quad (7)$$

where z is the number of immediate children of e_i^t . The mediatability of the photo element in the target schema in Figure 3(b) is the average mediatability of its children. Since all the child elements of the photo element have a mediatability of 0.25, the mediatability of the photo element is 0.25.

Before we define the mediatability between the source and target schemas, we make a small but important observation. Once the mediatabilities are computed for all elements, it is possible that the root element of the target

schema has more than one member in its similarity set. This implies that the source schema may have more than one substructure that can be mediated with the target schema. Identifying the correct substructure would involve considerable effort from the developer. To reflect this, we take the cardinality of the similarity set of the root element in defining the mediatability between the target and source schemas. We now define the mediatability between the target and source schemas as the ratio of the mediatability of the root element of the target schema and the cardinality of its mediatable set.

$$\sigma(H_t, H_s) = \frac{1}{|S(\text{root of } H_t)|} \sigma(\text{root of } H_t) \quad (8)$$

The mediatability between the two schemas in Figure 3 is computed as follows. The mediatability of the root element (SearchResult) is 0.25. The similarity set of the SearchResult element, which is the root, is {Result}. The cardinality of the similarity set of the root is 1 and its mediatability is 0.25. Computing the mediatability between the two schemas as defined in Equation 8, we get 0.25.

3.6 Optimizing Time Complexity

One of the drawbacks of the approach to comparing every element in the target schema is that the computational complexity is $O(n^2)$. This inefficiency further enhanced by the fact that often times comparison will yield no meaningful results. As a way of optimizing this comparison, we define the scope of comparison. We adopt a method similar to $\alpha\beta$ pruning to reduce the number of elements in the source schema that need to be compared with a given element in the target schema. The children of an element e_i^t in the target schema would be compared only with the children of those elements in the similarity set of e_i^t . The children of those elements in the source schema that belong to the similarity set of e_i^t are the scope of comparison for the children of e_i^t . Defining the scope of comparison would help reduce the complexity of the average running time of element similarity computation. In our example, the width element in the target schema would be compared with the children of the image element in the source schema, since the image element in source schema A is in the mediatability similarity set of the parent element of width.

4 Evaluation

In this section we present the empirical evaluations of our algorithm. The objective of our empirical evaluations is three fold: 1. Evaluate the accuracy of our approach through a user study; 2. Study the impact of semantic annotation on mediatability and 3. Demonstrate the scalability of our algorithm.

In our experiments, we compare a target schema with 5 different source schemas. The source schemas are created by studying the results schemas of Yahoo Web Search⁶(schema A), Google Search⁷(schema B), Microsoft Live Search⁸(schema C), Yahoo Image Search⁹(schema D) and Flickr¹⁰ (schema E).The schemas for Google Web Search and Flickr search were created by studying their responses, since they do not provide XML schemas explicitly. The experimental datasets including the schemas and the ontologies that are used in annotation, the user study questionnaire and an implementation of the mediatability computation algorithm are available at [1].

In our experiments, subclass similarity is assigned a value of 0.5 and superclass similarity is assigned a value of 0.8. Hyponym and hypernym scores are calculated as $\frac{1}{l}$, where l is the length of the hyponym or hypernym relationship in wordnet. The Levenshtein measure is used in the computation of lexical similarity.

4.1 Evaluating Accuracy

Our first experiment compares the mediatability scores obtained by our algorithm with a set of normal and expert users. The set of expert users comprised of committers of XML centric Apache projects including Apache Axis and Apache XML Schema. Normal users consisted of mashup developers having minimal programming and XML expertise. We included the normal users to compare our scores with the perceived difficulty of average developers, who we believe will have the most benefit from our work. Users were asked to rank the mediatability between the source and the target schemas using a Web application. Our results, illustrated in Figure 4, show that the calculated mediatability scores match fairly well with the perceived mediatability values and agree well with the expert opinions. The average margin of error between the system calculated mediatability and the perceived mediatability of the normal users was less than 15%, while the margin of error with expert uses was less than 10%. We make a special observation about schemas A and E. We recall here that schema A was derived from Yahoo Web Search. This schema did not have any image element in its result set and hence was given a low mediatability score to account for the loss of information. However, users perceived the mediatability to be twice as easier than the system calculated value. This indicates that our approach is very conservative and does not over-estimate. Similarly, schema E (derived from Flickr), had a structural heterogeneity, that was penalized by the system.

⁶<http://developer.yahoo.com/search/Web/V1/WebSearch.html>

⁷<http://code.google.com/apis/ajaxsearch/>

⁸<http://dev.live.com/livesearch/>

⁹<http://developer.yahoo.com/search/image/V1/imageSearch.html>

¹⁰<http://www.flickr.com/services/api/flickr.photos.search.html>

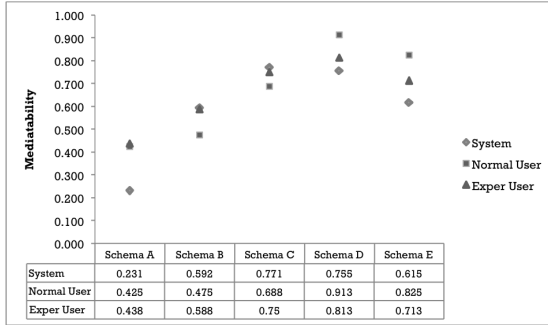


Figure 4. Accuracy Based on User Study

4.2 Impact of Annotation

This experiment measures the impact of semantic annotations in determining the mediatability. We annotated the source and target schemas with concepts from the semre descriptor ontology[1], a categorization of Web API's derived from ProgrammableWeb. The mediatability was calculated when the schemas have no annotations, partial annotations and complete annotations. The schemas were annotated using the techniques described in the SAWSDL recommendation [18]. Schemas with partial annotations were created by adding annotations top-level annotations to complex types. Schemas with complete annotations were created by adding annotations to the leaf elements in addition to the top-level annotations. Figure 5 illustrates the impact of annotation on mediatability. In the case of schema A, where the target schema has more elements than the source schema, the mediatability is low in all the three cases. However, we can see that semantic annotations considerably improve the mediatability score. Having partial annotations does not impact the mediatability in the case of schema A, since there are no complex types in the source schema. In the case of schemas B, C and D that contain complex types, one can see that complete annotations significantly improves the mediatability score and even partial annotations have an impact on the mediatability. On average our experiments demonstrate that partial annotations improve the mediatability by a factor of 2 while having complete annotations improves the mediatability by a factor of 3.

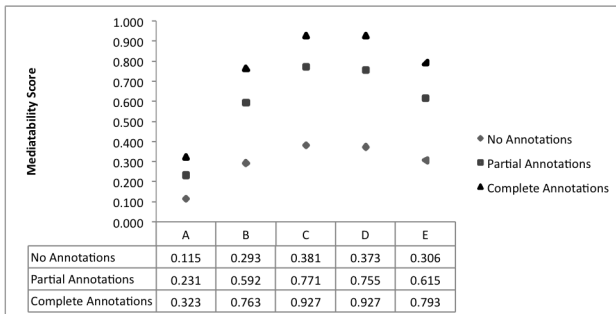


Figure 5. Impact of Semantic Annotation

4.3 Evaluating the Scalability

Our third experiment demonstrates the scalability of our algorithm. The algorithm was tested on two systems with different computing resources. System 1 is a Mac Book Pro running OSX 10.5 with 2 GB RAM and Intel Dual Core 2.0 GHz processor. System 2 is a Dell server running Fedora Core 5 with 16 GB RAM and AMD Quad Core 2.4 GHz processor. As illustrated in Figure 6, we see that in the

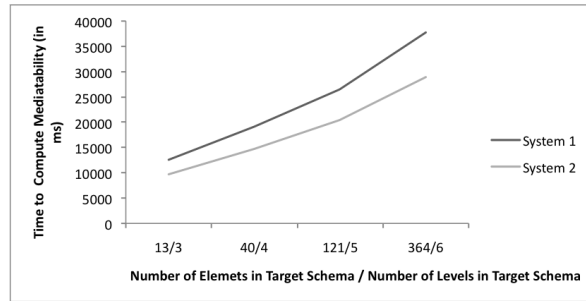


Figure 6. Measuring Execution Time

worst case, system 1 takes 36 seconds to compute the mediatability and system 2 accomplished the task in 25 seconds. This demonstrates the scalability of our algorithm. Figure 6 measures the scalability when the source schema has 364 elements and is 6 levels deep and the number of elements in the target schema are varied from 13 to 364. The depth of the target schema was varied from 3 to 6.

5 Related Work

The primary focus of this paper is to define a computable metric for measuring the ease of mediating between two schemas. The research presented in this paper is inspired by and builds upon the past work in the areas of database, XML and ontology schema matching. We believe that to the best of our knowledge, there has not been any previous research to estimate the degree and complexity of human involvement in XML schema mediation.

Since the early work on federated databases [20], interoperability among databases with heterogeneous schemas has been a well researched issue. Research in the area of database schema integration like [14]and [8] discuss approaches to matching that transform heterogeneous models into a common model. [17] discusses an approach for automatic annotation by converting XML descriptions to schema graphs to facilitate better matching. [10] abstracts the mappings between models as high level operations independent of the underlying data model and the applications of interest. [11] discuss an approach to computing the matching between two schemas based on similarity flooding. The approach presented in [11] computes the similarity of an element, based on the similarity of the neighboring elements in a graph.

The various heterogeneities that can exist between two schemas is discussed in [7]. This is further extended in the context of Web services, where message level heterogeneities between two interoperating Web services are studied in detail [16].

In the area of semantic Web services, the WSMO project [2] which coined the term *Data Meditation*, is most relevant to our work. Much of the focus of WSMO research has been in ontology mapping. [4] discusses an mediator based approach to address data and process mediation. [15] present a formal model for ontology mapping. [15] further discusses the role of the formal model in creating and expressing mappings in WSML, based on semantic relationships. [21] discusses an integrated model based on data level, functional level and process mediation for the Semantic Web with the main focus on services created using WSMO. Ontology matching and mapping is a vast area of research. In addition to the WSMO approach to ontology mediation, [3] and [12] among others also address this problem in different contexts. However, as discussed before, the measure of difficulty in data mediation (as captured by mediatability) and comprehensive evaluation with real world data as presented in this paper is missing.

6 Conclusion and Future Work

In this paper we introduce the concept of mediatability as an indicator of the ease of mediation between two schemas. We also provide a quantifiable and computable definition for mediatability. We present a simple two pass algorithm for computing mediatability between two schemas. The first pass of the algorithm computes the element and the structural similarity and the mediatability is computed in the second pass. We adopt a pruning strategy based on $\alpha\beta$ pruning to improve the average case performance of our algorithm. Our experiments analyze the impact of having semantic annotations in determining the ease of mediation between two schemas. We validate our approach by comparing the mediatability scores generated by our system against that of user perceived difficulty in mediation and study the scalability of our algorithm in both client and server contexts.

While structural and element similarity are essential for computing mediatability, they are by no means sufficient. In this work, we do not consider the nullability and the cardinality properties of XML schema, that play a significant role in instance level mediation. Another interesting aspect to study would be the impact of various schema, element and attribute level heterogeneities discussed in [16] on the mediatability between two schemas. Finally, we propose to extend our work by addressing the relevance of mediatability to various automatic and semi-automatic schema matching approaches.

References

- [1] Mediatability Web Resource. <http://apihut.com/mediatability/>.
- [2] Web Services Modeling Ontology. <http://wsmo.org>.
- [3] D. Calvanese, G. D. Giacomo, and M. Lenzerini. Ontology of integration and integration of ontologies. In *Description Logics*, 2001.
- [4] E. Cimpian, A. Mocan, and M. Stollberg. Mediation enabled semantic web services usage. In *ASWC*, pages 459–473, 2006.
- [5] R. T. Fielding. *Architectural Styles and Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, UNIVERSITY OF CALIFORNIA, IRVINE, 2000.
- [6] D. Harel and R. E. Tarjan. Fast algorithms for finding nearest common ancestors. *SIAM J. Comput.*, 13(2):338–355, 1984.
- [7] V. Kahsyap and A. P. Sheth. Semantic and schematic similarities between database objects: a context-based approach. *VLDB Journal*, 1996.
- [8] J. Madhavan, P. A. Bernstein, and E. Rahm. Generic schema matching with cupid. In *VLDB*, pages 49–58, 2001.
- [9] E. M. Maximilien and A. Ranabahu. Ibm sharable code. <http://services.alphaworks.ibm.com/isc/>.
- [10] S. Melnik. *Generic Model Management: Concepts and Algorithms*, volume 2967 of *LNCS*. Springer, 2004.
- [11] S. Melnik, H. Garcia-Molina, and E. Rahm. Similarity flooding: a versatile graph matching algorithm and its application to schema matching. In *ICDE*, pages 117–128, 2002.
- [12] E. Mena, A. Illarramendi, V. Kashyap, and A. P. Sheth. Observer: An approach for query processing in global information systems based on interoperation across pre-existing ontologies. *Distr. and Parallel Databases*, 8(2):223–271, 2000.
- [13] G. A. Miller. Wordnet: A lexical database for english. In *HLT*, 1994.
- [14] R. J. Miller, M. A. Hernández, L. M. Haas, L.-L. Yan, C. T. H. Ho, R. Fagin, and L. Popa. The clio project: Managing heterogeneity. *SIGMOD Record*, 30(1):78–83, 2001.
- [15] A. Mocan, E. Cimpian, and M. Kerrigan. Formal model for ontology mapping creation. In *ISWC*, pages 459–472, 2006.
- [16] M. Nagarajan, K. Verma, A. P. Sheth, and J. A. Miller. Ontology driven data mediation in web services. *J. Web Services Research*, 2007.
- [17] A. A. Patil, S. A. Oundhakar, A. P. Sheth, and K. Verma. Meteor-s web service annotation framework. In *WWW*, pages 553–562, 2004.
- [18] SAWSDL Working Group. Semantic annotations for wsdl and xml schema.
- [19] A. P. Sheth, K. Gomadam, and J. Lathem. Sa-rest: Semantically interoperable and easier-to-use services and mashups. *IEEE Internet Computing*, 11(6):91–94, 2007.
- [20] A. P. Sheth and J. A. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Comput. Surv.*, 22(3):183–236, 1990.
- [21] M. Stollberg, E. Cimpian, A. Mocan, and D. Fensel. A semantic web mediation architecture. In *CSWWS*, pages 3–22, 2006.
- [22] Yahoo! Inc. Flickr: Explore everyone’s photos on a map. <http://www.flickr.com/map/>, 2008.