XML-BASED DATA EXCHANGE IN THE HETEROGENEOUS DATABASES (XDEHD)

Husam Ahmed Al Hamad
Department of Information Technology, College of Computer,
Qassim University, Qassim, Saudi Arabia,

ABSTRACT

While the world is witnessing an information revolution unprecedented and great speed in the growth of databases in all aspects. Databases interconnect with their content and schema but use different elements and structures to express the same concepts and relations, which may cause semantic and structural conflicts. This paper proposes a new technique for integration the heterogeneous eXtensible Markup Language (XML) schemas, under the name XDEHD. The returned mediated schema contains all concepts and relations of the sources without duplication. Detailed technique divides into three steps; First, extract all subschemas from the sources by decompose the schemas sources, each subschema contains three levels, these levels are ancestor, root and leaf. Thereafter, second, the technique matches and compares the subschemas and return the related candidate subschemas, semantic closeness function is implemented to measures the degree how similar the concepts of subschemas are modelled in the sources. Finally, create the medicate schema by integration the candidate subschemas, and then obtain the minimal and complete unified schema, association strength function is developed to compute closely of pair in candidate subschema across all data sources, and elements repetition function is employed to calculate how many times each element repeated between the candidate subschema.

KEYWORDS

Schema integration, Data matching, XML schema, Heterogeneous data sources, Mediated schema.

1. INTRODUCTION

The overgrowth of the internet and exchange the information have caused to increase the need for a common data format. The reason that gave rise to the need for a highly standardized common data format for data exchange and integrating between heterogeneous applications and systems [1], are the advantages of interoperability, exemplified by the web [2]. The database that obtained from the scientific experiments are mostly heterogeneous and distributed [3]. XML is one such innovative usage of relational database prompted by increasing the usage of organizations database applications and its related need of managing frequent storage and retrieval of not-very structured data in document format [4]. XML is designed to represent data using tags (elements), allows expressing information in ways that match better for business. XML allows us to model information systems in natural and intuitive way. It brings a number of powerful capabilities to information modeling such as heterogeneity, extensibility, and flexibility. For these reasons, XML becomes a standard data format widely used in these organizations and a common language for data transmission over the Internet. This leads to a growing need for exchanging and integrating the heterogonous XML data sources and schemas between different application systems. Therefore, data exchange between heterogeneous databases becomes very interesting topic recently. Numerous of languages such as Document Type Definition (DTD) is also used for restructuring the XML documents [5, 6], eXtensible Stylesheet Language Transformations (XSLT) can be used for creating a mediate architecture of XML schemas [7] as well.

XML representation may inefficient compared to storage the data in a relational database, since XML element (tag) names are repeated throughout the document. However, XML representation has significant advantages when use it to exchange the data between the organizations, and for storing complex structured information in files. XML makes the message self-documenting by presenting of the elements, a schema does not need expert to understand the meaning of the text. The format of XML document is not rigid; easily can add additional information using the elements. In addition, can ignore any information or element. In other words, the ability to recognize and ignore unexpected elements allows the format of the data to evolve over time, without invalidating existing applications. Similarly, the ability to have multiple occurrences of the same element makes it easy to represent multivalued attributes. Likewise, XML allows nested structures, and a wide variety of tools are available to assist in XML processing, including programming language to create and to read XML data, browser software, and database tools [8]. Many researchers [9, 10] investigate and study transformation and conversation theories for XML schema and relational model. Others [6] design and implement an interactive tool for data exchange between heterogeneous systems. Semantic conflicts in heterogeneous database systems has studied in many researches. Rajeswari and Varughese [11] divides the conflicts between values, attributes, and tables. The data exchange and metadata schema management can use for mapping between relational schemas as well [12]. Many problems such as message losing, relationship misjudging and field attribute changing during exchanging and conversion of heterogeneous data can be solve [13].

Using a mediated schema for a pair of heterogeneous data and a set of initial correspondences between attributes [14, 15] assist including elements and relationships in the mediated schema. A mediated schema integration approach for XML structures [16] adopts a pattern-growth structure mining approach [17] to reconcile a number of XML structures. Combines results of multiple matching algorithms to produce semantic correspondence between elements COMA/COMA++ [18, 19, 20] and combines results of multiple matching algorithms to produce semantic correspondence between elements. Self-configuring schema integration system based on the notion of probabilistic mediated schema, which is a set of mediated schemas associated with probabilities [21]. A mediated XML schema matching approach using paths with the input schemas encoded as trees, the approach defines a set of classifiers to measure various schema characteristics such as labels and path length QMatch [22].

2. PROBLEM SPECIFICATION

Traditional integration methods of heterogeneous XML schemas typically generate a mediated schema which is either complete or minimal [16] but not both. Conversely, the proposed technique in this research creates a single comprehensive mediated schema that holds both criteria complete and minimal. To do so, the technique resolve structural and semantic conflicts and put them in the proposed mediated schema.

Two challenges cause structural and semantic conflicts for heterogeneity of schema. First, structural conflicts appear when express the same relations by different XML structures, for example, two different paths of the same semantic. Second, similarly, semantic conflicts appear when different sources describe the same concepts using different element names, or there is a meaning overlap between similar concepts in different sources; for example, in

Figure 1 the "BloodTest" and "TBlood" refers the same domain "Blood Test". Therefore, when increase the number of sources, structural and semantic heterogeneity problem could become much worse [23].

There is also an integration difficulty between variant XML sources, all sources have a common domain but each has written for different purpose, even if they contain a set of semantic correspondences between their elements and structures for each. Different element labels may refer to similar concepts, therefore, weight uses for correspondence with a similarity score, for example in

Figure 2 (BloodTest ≈ TBlood, 0.95) which is mean the element BloodTest has 95% correspondence with TBlood. Likewise, there are different structures but refer to similar semantic; for example, Patient\Department and Department\Patient show a similar association between Patient and Department. The proposed technique in this paper will first gives a set of weights for these inputs to solve any conflict, and then achieve the comparison.

The proposed technique resolves many conflicts and difficulties, it seeks to create a comprehensive (complete) mediated schema between a set of heterogonous databases by integration the XML schemas sources. The mediated schema assists to access data from different sources effectively and efficiently; to achieve this goal, the mediated schema should satisfy two criteria: schema completeness and schema minimality. The technique calculates the complete and minimal schema by matching between the levels of original schemas. Complete schema means the calculated mediate schema preserves all of the relations of the sources. Minimal schema means that each of these relations appears only once, without redundancy. Fully, the final calculated mediate schema preserves all information (relations and contents) from the sources in a minimal form.

3. TECHNIQUE DETAILS

Proposed method in this research introduces a new technique for refine a comprehensive mediated schema that effectively integrates multiple heterogeneous XML sources. The approach favors the integration of relations over that of separate concepts because the relations carry domain information. Moreover, integrate a relation will naturally integrate the concepts embedded within that relation, but the reverse might not be true. In the context of this paper, the completeness measure refers to the extent to which both concepts and relations are preserved.
The technique develop and use three different inputs, First input is a set of XML sources, which includes XML schemas and their instances. Second input is a set of semantic correspondences between individual elements, which can be provided, for example elements e1 and e2 are attached with a similarity *escore s(e1, e2) ∈ [0, 1]*. Finally, third input is a set of weights provides by domain experts such as an input file, the aim of weights is combine similarity scores between elements or structures, and select suitable candidates for creating the final schema (Lee et al. 2007) [24]. The technique combines automatically between the elements and structures of the heterogeneous sources by choose the prevalent structure among the sources. The technique integrates the selected relations into the proposed mediated schema by captures the most domain concepts and structures.

A comprehensive mediated schema that contains relations sources data increases the effectiveness of returning correct answers to a user query [23]. It is worth noting that preserving all relations implies preserving all of the concepts that are associated with these relations. In other words, all concepts in the sources are also preserved. In our context, the completeness (C) of a final schema is formally defined as the percentage of the final schema's relations, which is can be found in the sources R. The percentage of completeness (C) = G / R, where G is total relations of global mediated schema, and R is total relations of the sources.

A minimal mediated schema describes all relevant domain relations only once. The minimal representation helps reducing the number of search operations for a user query over the mediated schema, thus increasing the efficiency of data access. Schema minimality measures the extent to which the mediated schema is compactly modeled without redundancies, the percentage of minimality (M) = 1 - (RG / G), where, RG is redundant relations of G, and G is total relations of global mediated schema.

The mediate schema means a schema that contains structure and context of the heterogeneous schemas, which is complete and minimal. The main steps for creating a comprehensive mediate schema of heterogeneous sources are proceed through three points: (i) extract subschemas, (ii) match subschemas, and finally (iii) cluster and create the final medicate schema.

Figure 1 illustrates two sample of XML sources.

Figure 2 displays the tree structure of the XML source after mapping.

3.1. Extract subschemas

Extracting aims to prepare the sources and convert them to subschemas, extracting output is used as an input for matching step. To extract the set of subschemas from the sources, degree three is employed to represent the maximum length of the structure path; this degree contains three levels for any path (ancestor, root, and leaf). Degree means number of levels that will include in each subschemas; for example, ancestor, root and leaf levels mention to degree three, degree four contains additional level and so on. Degree three considers a high value to output a minimal mediate schema. For parent–child paths at that degree, non-leaf elements has fewer redundancies because many non-leaf elements obtain relevant elements for the mediated schema. Each subschema contains at most three elements extracted from the sources based on degree three.
Based on the previous two-tree structure of XML sources, the step here extract all sources subschemas of each, the technique extracts the first three levels (degree three) for each subschema as shown in

Figure 3. First element is the ancestor, second element is the root, and third element is the leaf, where the first level is the root, therefore the ancestor considers not exist at that level. For example, the extracting result of subschema number 1 from the first source is the root (Patient) and leaf (Details) without any ancestor because the firs node of the source considers a root, then extract the rest subschemas of level 1, subschema result is Patient\Details. After complete extracting the first level, the technique moves for extracting the next level (level 2) using the same rule. For example, subschema number 4 from the first source contains the ancestor (Patient), root (Details), and leaf (ID), subschema result is Patient\Details\ID, these steps are repeated until extract all subschemas of the sources as shown in

Figure 3

```
<Patient>
   <Details>
      <ID>125456</ ID>
      <Name>Sara Khalid</ Name>
      <Phone>58451514</ Phone >
   </ Details>
   <Treatment>
      <Test>
         <Xray>Negative</ Xray>
```

```
<Department>
   <Phone>6254236</ Phone>
   <Patient>
      <Phone>54515454</ Phone >
      <PName>Ibraheem Saeed</ PName>
      <NID>545415584</ NID>
      <BloodT>
         <CBE>AB+</ CBC>
         <Plates>290</ Plates>
```

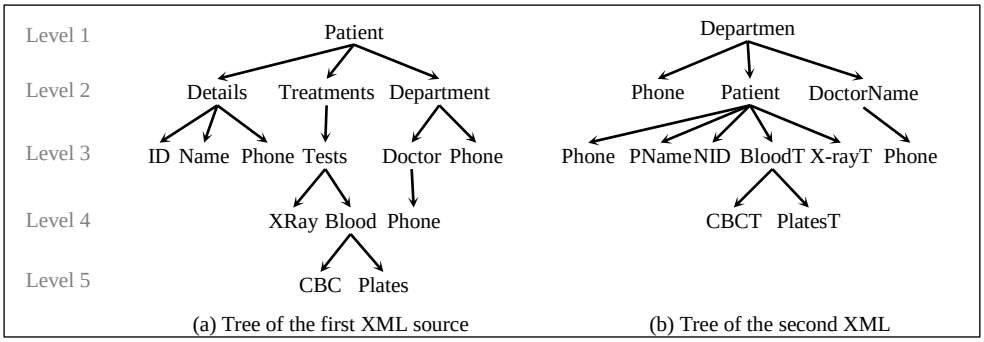
```
          <Blood>
             <CBE>A+</ CBC>
             <Plates>251</ Plates>
          </ Blood>
       </ Test >
    </ Treatment>
    <Department>
       <Doctor>
          <Phone>658542</ Phone>
       </ Doctor >
       <Phone>152145</ Phone>
    </ Department>
</ Patient>
```

(a) First XML source

```
       </ BloodT>
       <X-rayT>545415584</ X-rayT>
    </ Patient>
    <DoctorName>
       <Phone>545454565</ Phone>
    </ Doctor Name>
</ Department>
```

(b) Second XML source

Figure 1. Two samples of XML sources.



(a) Tree of the first XML source (b) Tree of the second XML

Figure 2. Tree structure of the XML sources.

Sub-schema	Root level	S1L1 (ancestor)	S1L2 (root)	S1L3 (leaf)
1	1	-	Patient	Details
2	1	-	Patient	Treatments
3	1	-	Patient	Department
4	2	Patient	Details	ID
5	2	Patient	Details	Name
6	2	Patient	Details	Phone
7	2	Patient	Treatments	Tests
8	2	Patient	Department	Doctor
9	2	Patient	Department	Phone
10	3	Treatments	Tests	XRay
11	3	Treatments	Tests	Blood
12	4	Tests	Blood	CBC
13	4	Tests	Blood	Plates
14	4	Department	Doctor	Phone

(a) Subschema of the first XML Source

Sub-schema	Root level	S2L1 (ancestor)	S2L2 (root)	S2L3 (leaf)
1	1	-	Department	Phone
2	1	-	Department	Patient
3	1	-	Department	DoctorName
4	2	Department	Patient	Phone
5	2	Department	Patient	PName
6	2	Department	Patient	NID
7	2	Department	Patient	BloodT
8	2	Department	Patient	X-rayT
9	2	Department	Doctor Name	Phone
10	3	Patient	BloodT	CBCT
11	3	Patient	BloodT	PlatesT

(b) Subschema of the second XML Source

Figure 3. Extracting subschemas of two XML sources

The *SubschemaExtracting* algorithm in List 1 presents pseudo-code for extracting the subschemas of data sources. The algorithm starts define a number of sources which is equal 2, then initializes a variable for saving the extracted subschemas results. Nested 'For' clause is used for calculating all subschema paths. In order for these subschemas to be included in the candidate list of subschemas, matching between these subschemas across all data sources are made. Such a matching resolution will be done in the next step.

List 1. Subschema extracting algorithm

```
SubschemaExtracting (S: set of source,
        Sub: set of subschema,
        R: roots, l: leaves)
n = 2 /*number of sources are 2*/
Subi ← {} /*Initialize set of subschemas*/
for each sources s (i <= n) loop
  for each root r loop
     for each leaf l loop
       if root and leaf are set then
          Subi ← {ancestor, root, leaf}
Return Subi
```

3.2. Match subschemas

Matching step enables obtain a set of semantic mappings between subschemas of all sources. In this step, the elements of subschemas that have extracted by the previous step is matched; each subschema contains set of elements. Matching mechanism compares first subschema of first source with all other subschemas of the second source, the compression considers the context structural of subschemas through closely matching of parent–child paths or ancestor-root-leaf.

Figure 4 shows matching steps between subschemas of the sources. To achieve the matching, the given correspondences values between the structures of subschemas are employed, these values determines the result of matching between the subschemas and their levels, the correspondences values that related to the elements is similarity to score exceed a given weight.

Matching Step	Degree Number	Source 1				Source 2		
		S_1L_1	S_1L_2	S_1L_3		S_2L_1	S_2L_2	S_2L_3
1	3	⌊	⌊	⌋		⌊	⌊	⌋
2	2	⌊	⌋			⌊	⌋	
3	2	⌊	⌋				⌊	⌋
4	2	⌊	⌋			⌊		⌋
5	2		⌊	⌋		⌊	⌋	
6	2		⌊	⌋			⌊	⌋
7	2		⌊	⌋		⌊		⌋
8	2	⌊		⌋		⌊	⌋	
9	2	⌊		⌋			⌊	⌋
10	2	⌊		⌋		⌊		⌋
11	1	|				|	|	|
12	1		|			|	|	|
13	1			|		|	|	|

Figure 4. Matching steps between subschemas of the sources

After this, the candidate pairs of matched subschemas between sources data are found. If two concepts in a relation are structurally close, this leads to consider them more likely to be related to each other. For example, in

Figure 2 the two concepts Department, DoctorName and Phone (number 14 in the first source with number 9 in the second source) are closely connected through parent–child paths or ancestor-root-leaf or any selected part in all sources, so they have a stronger semantic relation than the pair of concepts store.

Relation semantic closeness function [23] is measure the extent how well the elements in the sources represent the concepts of a relation. In other words, measure the similarity degree of

subschemas in the sources. Consider a relation R = x//y, where x and y are two concepts, each of which is a matcher of similar elements. If each of these elements concepts are highly similar, then these elements are more likely to represent the concept correctly. For example, the concepts "Phone" or "Patient" are labeled with exactly the same element name in all schemas as shown in

Figure 1 and

Figure 2, while element "Name" is arbitrarily labeled in the schemas: "DoctorName" and "PName". Intuitively, more confidence in concluding that model is more likely to refer to the same concept than element name.

Relation semantic closeness calculates by given weights w1 and w2 for each element, the closeness of a relation R = x//y is defined as *SemanticCloseness*(x//y) = w1 × escore(x) + w2 × escore(y), where escore(x) is the average similarity score of all of the source elements representing the concept x; the same for escore(y); and w1 + w2 = 1. The weights w1 and w2 indicate to the similarity significance of the ancestor concept x and descendant concept y to the relation R. Ancestors are typically given higher weights than descendants (i.e., w1 ≥ w2) because elements closer to the root are usually assumed to be more semantically important to the domain. In this research, the assumption holds when the descendant concept is a leaf and the ancestor concept is a non-leaf. However, the technique suggests that the two weights should be set equal by default (i.e., w1 = w2 = 0.5), especially when both concepts are non-leaves, because their roles should make equivalent impacts on the domain when the number of sources increases. For example, the two concepts BloodT and Blood have similar roles in the two candidate relations Patient//Blood and Patient//BloodT that convey equivalent meanings in the domain.

The technique matches three levels for comprising the similar structure and semantic, matching starts with degree three, which is the maximum length or the highest degree. Therefore, the initial value of degree *i* in this case will equal the highest degree *n*, $d_n=d_i$ where $n=i=3$. This means the comparisons include all levels of each subschema together as one block. As shown in

Figure 3, matching number 1 represents the degree number 3, the comparison between the first schema S1 that includes the levels (L) numbers 1, 2, 3 and with the second schema S2 that includes the levels numbers 1, 2, 3. In other words, matching is between S1L1, S1L2, S1L3 and S2L1, S2L2, S2L3 as one piece at the same order.

After comparison all levels in degree three, the new comparison degree is number two (new degree $d=d_{i-1} \Rightarrow d=d_{3-1} \Rightarrow d=d_2$), this means the compression between the levels will be in the form of pairs. After the comparison and as seen in

Figure 3, there is no matching in subschemas number 2, 3, 4, 5, 7, 10, 11, and 12 but there is matching in subschema number 6 between S1L2, S1L3 and S2L2, S2L3.

Figure 5.b and

Figure 5.c show the results of these matching. Likewise, there is matching in number 8 between S1L1, S1L3 and S2L1, S2L2, but because ancestor level is not contain any value then can ignore this matching, where it will not benefit later, the fact that one of the elements (ancestor) of the comparison is empty. Also, there is matching in number 9 between S1L1, S1L3 and S2L2, S2L3.

Figure 5.d shows the results of this matching. Likewise, there is matching in number 13 between S1L3 and S2L3.

Figure 5.e shows the results of this matching. As shown in

Figure 3, the remain subschemas 1, 2, and 7 form the first schema are still remain. If there is no value in ancestor, and other levels already matched before, then it is better to ignore eliminate it, this rule is applied in subschemas number 1 and 2. In addition, if the leaf of the subschema represents a root in the original schema, then it is better also to ignore and not count it, this rule applies to subschema number 7.

In List 2, the *Matching* algorithm presents pseudo-code to find the candidate subschemas across all data sources. The algorithm begins by initialize a variable to save the candidates subschemas based on the technique that mentioned before. Nested 'For' clause is used for calculating the path for every candidate subschema of the two data sources and included it in the final mediated schema, if the subschema relation path of the first source is closeness to subschema path of the second source then the algorithm considers the subschema a candidate subschema. Thereafter, delete the primary subschemas of the sources from the list of subschemas and then complete the loop to find all candidates. The technique needs to make clustering between these candidate subschemas across all data sources. Such a clustering resolution will be done in the next step.

List 2. Subschemas matching algorithm

```
Matching (S1: set of source1, S2: set of source2, Sub1: set of
        subschema1, Sub2: set of subschema2, )
CandidatesSub ← {} /*Initialize set of matching*/
d =3 /*degree of subschema is 3*/
calculate SemanticCloseness of sub1 and sbu2
for each (j=d, j >=1, j--) loop
  for each subschema in S1 loop
     for each subschema in S2 loop
       if Sub1(x//y) semantic closeness Sub2(x//y) then
         Add Sub1, Sub2 into CandidatesSub
         Remove Sub1, Sub2 from Sources S1 and S2
Return CandidatesSub
```

As shown in

Figure 5, there are five nominated concepts sets could match. Based on our technique, the best one from each candidate pair is that rounded by dashes squares.

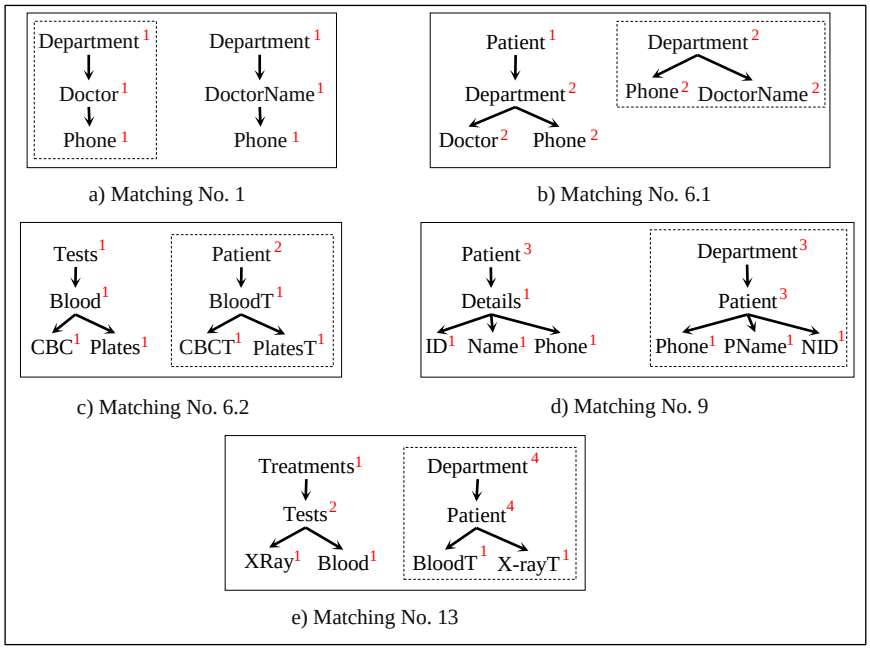


Figure 5. Candidate and nominated subschemas of all sources

3.3. Cluster and create the medicate schema

After matching, that retained information values of all sources leaves and organized them as candidate subschemas. In this step, the technique defines how subschemas can be semantically equivalent, and then creates the target schema by clustering and integrating the subschemas together based on association strength and relation semantic closeness, then finally, constructs the final mediate schema of the sources which it contains structure and information of the sources. The subschemas results at the last step consider candidate subschemas, one subschema structure from each nominated pair in

Figure 5 considers valid structure, and then include the nominated subschema into the final mediate schema.

Association strength function is defined to compute how closely the pair of concepts in candidate subschema are associated within their relations across all data sources. Association strength calculates all possibilities of the path length of its corresponding paths. The matching works when path length ≤ three, it starts with the degree three then two then finally with degree one. The path length function length(x//y) returns the number of elements traversing from x to y. Since path x//y appears in each subschema, the length is equal and greater than one and equal and less than three ($1 \leq \text{length}(x//y) \leq 3$). Therefore, minimum length is equal one, and maximum length is equal three. The equation of association is ass(x//y) = 1/ length(x//y), and the association strength between two paths is defined as *AssociationStrength*=max[ass(R1), ass(R2)]. For example, suppose R1 = Department//Doctor, and R2 = Department//DoctorName, their length are length(R1) = 1 and length(R2) = 1, then the associations are ass(R1) = 1/1=1 and ass(R2) = 1/1=1. Both values are normalized between 0 and 1; therefore, this means both of them semantically related to together. Another example, if R1 = Patient//XRay which appears one time and R2 = patient//X-rayT which appears also one time, their length are length(R1) = 3 and length(R2) = 1, and their association are ass(R1) = 1/3=0.33 and ass(R2) = 1/1=1. So, because ass(R2) > ass(R1), this means X-rayT is more semantically related to Patient than Xray.

Mediate schema produces best candidate subschemas pair. Association strength function is employed to find the suitable roots or non-leaves concepts; the suitable root means choose all candidates that are associated with leaf concepts then attached structure of all sources. Element repetition function of leaf and non-leaf is calculated for each candidate subschemas pair. Repetition function calculates cumulatively the frequency of each element in the candidate subschema during the matching process. Therefore, if the first level (root) has a highest repetition then its subschema will select to be part of the final medicate schema. If the first level of each pair are same, then the second level will consider the measure of clustering and so on. The highest repetition degree of the elements is chosen because repetition degree considers closer to the last chosen subschema and usually assumed to be more semantically important to the domain. For example in

Figure 5, in matching (a) the selected subschema between the two candidates is one of them because both are same. In matching (b) the selected subschema is the second because the repetition of its first level (root) is greater than the other, the same for matching (c) and (e). In matching (d) where repetition of the first level of both are equal, then the second level considers the measure, therefore, and because the repetition of second level of the second subschema is greater than the first subschema then the selected subschema is the second.

The technique solve by default many structural conflicts such as nesting discrepancy, backward path and structural diversity (Nguyen et al., 2011) [23]. Nesting discrepancy refers to structural conflicts when a concept can be modeled as a descendant of another concept with different nesting levels, or path lengths, to convey similar semantics. For example in

Figure 2, the element "Name" ultimately refers to Patient regardless of being named as a parent–child path Patient//Name in the first source (a), or as an ancestor–descendant path Patient/PName in the second source (b). Backward paths may lead to structural conflicts because it can express the similar meanings using forward and backward paths as well as different hierarchical directions. In addition, both Patient//Name in the first source (a) and Department//Name in the second source (b) possess a similar meanings with opposite ancestor–descendant directions.

In List 3, the Clustering algorithm presents pseudo-code to find the final mediated schema across all candidate subschemas. The algorithm begins by initialize a variable to save the final medicate schema based on the technique that mentioned before. The algorithm calculates the path association strength for every candidate subschemas attained during the previous step. Whenever a conflict arises, the most qualified candidates are those that favor high path association. Thereafter calculate element repetition in each level for all candidate subschemas. If the two candidate subschemas are same then add the first or the second into the final mediate schema, else if the ancestor repetition of the first candidate subschemas is greater than ancestor repetition of the second candidate subschemas then add the first in to final mediate schema else the second. Else if the root repetition of the first candidate subschemas is greater than root repetition of the second candidate subschemas then add the first in to final mediate schema else the second. Else if the leaf repetition of the first candidate subschemas is greater than leaf repetition of the second candidate subschemas then add the first in to final mediate schema else the second. Eventually, return the final mediate schema.

List 3, Medicate Schema extracting algorithm.

```
Clustering (S1: set of source1, S2: set of source2, Sub1: set of
        subschema1, Sub2: set of subschema2, )
FinalMedicateSchema ← {} /*Initialize set of matching*/
```

```
calculate AssociationStrength of Sub1 and Sbu2
calculate ElementRepetition of Sub1 and Sbu2
for each Sub1 and Sub2 in CandidatesSub loop
  if Sub1 equivalents Sub2 then
     add Sub1 or Sub2 into MedicateSchema
  elseif 1st Level repetition and association Sub1 > 1st Level repetition and
     association Sub2 then
     add Sub1 into MedicateSchema
  else add Sub2 into MedicateSchema
  elseif 2nd Level repetition Sub1 and association > 2nd Level repetition and
     association Sub2 then
     add Sub1 into MedicateSchema
  else add Sub2 into MedicateSchema
  elseif 3rd Level repetition and association Sub1 > 3rd Level repetition and
     association Sub2 then
     add Sub1 into MedicateSchema
  else Add Sub2 into MedicateSchema
Return FinalMedicateSchema
```

The final mediate schema after integrating all returned candidate subschemas is shown in

Figure 6, the medicate layer preserve all of the information from the sources. The algorithm keeps all of the leaf concepts; and subsequently, the non-leaf concepts associated within their corresponding relations are preserved accordingly.

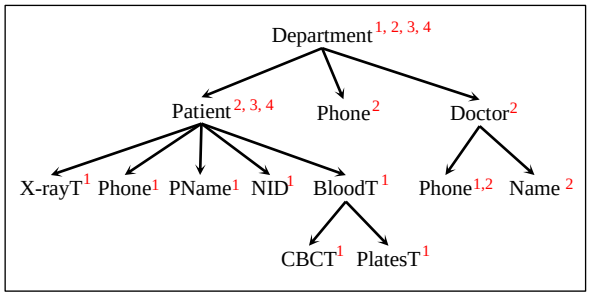


Figure 6. The final schema after integrate all sub-trees

4. EXPERIMENTAL RESULTS

In this section, the output quality of the mediated schema is evaluated. The output quality measures how well the mediated schema represents the source schemas via the semantic mappings. The mediated schema quality is measured using completion and schema minimality. The technique with four of the most recent integration databases are compared, these databases are PORSCHE [16], COMA++) [18, 19], RONDO [25] and XINTOR [23]. These databases are differentiate based on the number of schemas they handle. Only a pair of schemas is used at a same time, whereas the technique integrates only two schemas. Experiments are performed on the real dataset OAGIS [26] and synthetic datasets BOOKS [16] and MOVIES [27] with their characteristics in Table 1.

Table 1 Experimental schemas datasets [23]

Characteristics	OAGIS	MOVIES	BOOKS
Number of schema trees	108	1,312	176
Number of distinct labels	925	87	19

Total number of elements	218,762	64,706	1,320
Schema Size - Smallest	99	14	5
Schema Size - Largest	11,972	91	14
Schema Size - Average	2,025.57	49.32	7.59

The four databases are compared in the same manner: they all process two source schemas at a time. Thus, 10 pairs of schemas are randomly selected from the sources: four from MOVIES, three from BOOKS, and three from OAGIS. These pairs are the input of all techniques.

Regarding completion schema, the technique produces a mediate schema has high completion. As a comparison with other methods results, the result of our mediate schema has higher result than COMA++ in (M1, M2, M3, M4, O1, and O2). In addition, our result has almost higher result than RONDO in (M1, and M4). As well, the comparison with PORSCHE the result is higher in (M2, M4, and O2). Finally, the result is equal with XINTOR in (M1, and O1) but it is little lowest in (M2, M3, M4, O2). As shown in

Figure 7, the proposed technique in general produces a promise results compared with other methods. COMA++ does not match well when the same label appears multiple times in the schemas and at different nesting levels under the same ancestor element as in M2 and M4. In these cases, RONDO performs relatively well. PORSCHE can identify such nesting discrepancies but fails to discover the backward paths in M2. XINTOR performs well because it correctly discovers leaf elements to be associated with their relevant non-leaf elements in composite concepts. XDEHD is almost performs well and successes to output completion schemas than RONDO, PORSCHE, and COMA++ but not than XINTOR, it also at most cases successes discover and associate the leaf elements with their relevant non-leaf elements.

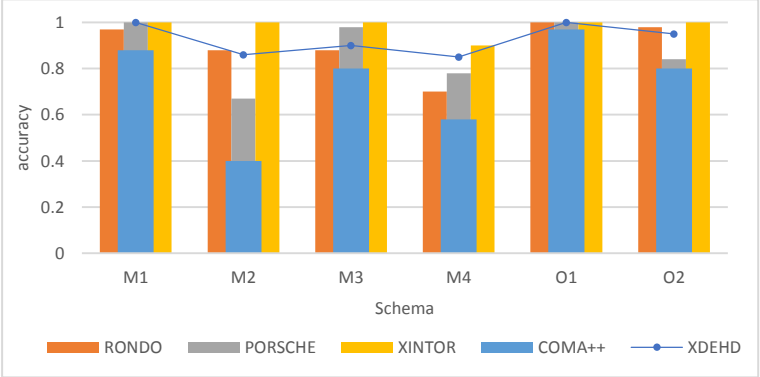


Figure 7. Schema completion of XINTOR, COMA++, RONDO, PORSCHE, and XDEHD.

The final mediate sachem has also achieve the minimality, when the maximum length is high (degree equal 3), the number of redundant elements is small due to a high degree of relevant pairs of elements. When is smaller than many paths become redundant,

Figure 8 illustrates the schema minimality of the algorithm at maximum length (degree) = 1, 2, and 3. For parent–child paths (maximum length = 1), non-leaf elements cause redundancies because many non-leaf elements which associated with a leaf element do not convey any meaning, the same resound when maximum length degree is equal 2 but with less impact. In this experiment, the mediate schema result can be view more favorably with the maximum length degree equal 3 because there are few redundancies while obtaining more relevant elements for the mediated schema.

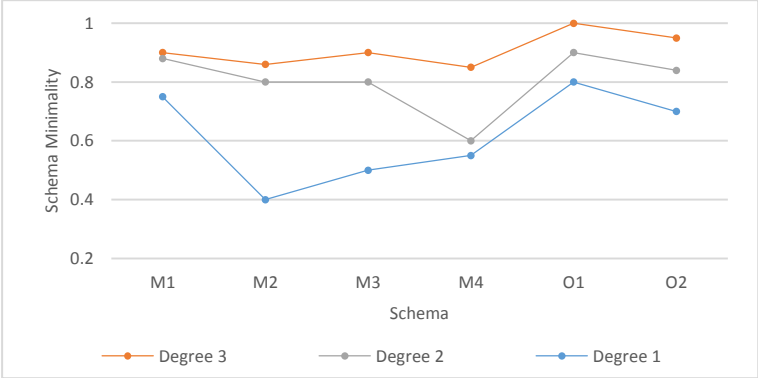


Figure 8. Schema minimality of the algorithm at maximum length (degree) = 1, 2, and 3.

5. CONCLUSION

This research was presented a new technique generates complete and minimal mediate schema integration between a set of heterogenic schema sources, it was complete and minimal. A simple method was proposed to decompose and extract the sources structure. Weight and relation semantic closeness function were developed to find the candidate subschemas. Further, the association strength and element repetition functions were augmented to configure the final mediate schema. The experiments demonstrated the efficiency and usefulness of the final mediated schema comparing with literature results. In future work, we are looking to enhance the method by address the semantic conflicts of overlapping concepts in source schemas.

ACKNOWLEDGMENT

The authors would like to acknowledge the financial support provided by the Deanship of Scientific Research at Qassim University, under research project reference number (1932/1434), entitled, "XML-Based Data Exchange in the Heterogeneous Database (XDEHD)".

REFERENCES:

[1] Juan D., Zheng, Q. (2012) "The research on the XML-based information exchange under heterogeneous Enviornment in HR Oursourcing enterprises", Computer Science & Education (ICCSE), 7th International Conference, 14-17 July 2012, pp.462-465.

[2] Papamarkos, G., Zamboulis L., Poulovassilis, A. (1998) XML Databases, School of Computer Science and Information Systems, Birkbeck College, University of London, http://www.dcs.bbk.ac.uk/~sven/adm08/xmlDBs.pdf

[3] Gancheva V., Shishedjiev B., Kalcheva-Yovkova E. (2011)"An approach to convert scientific data description", Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), IEEE 6th International Conference, 15-17 Sept. 2011, pp.564-568,

[4] Joseph D, "Current usage and future of XML Database Management Systems", 2009, http://www.getallarticles.com/2009/12/28/4/

[5] Arenas M, Barcelo P., Libkin L., and Synthesis F. M. (2010) "*Relational and XML Data Exchange*", Vol. 2(1), pp. 1-112.

[6] Arenas M., Libkin L. (2005) "Xml data exchange: Consistency and query answering" In Proceedings of the 24th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS'05), Baltimore, USA, 2005, pp. 13-24.

[7] Jumaa H., Rubel P., Fayn J., (2010) "An XML-based framework for automating data exchange in healthcare", e-Health Networking Applications and Services (Healthcom), 12th IEEE International Conference, 1-3 July 2010, pp.26124-269.

[8] Silberschatz A., Korth H. F., Sudarshan S. (2010) *Database System Concepts*, McGraw-Hill, 2010, Sixth Edition.

[9] Qian L. (2010) *Research and implementation of XML-based data transformation for heterogeneous Database*, Master's Thesis, Shenyang University of Technology, Mar. 2010.

[10] Wang S. "*Research of data exchange based on XML for heterogeneous data*", Master's Thesis, Harbin University of Science and Technology, Mar. 2008.

[11] Rajeswari V., Varughese D. K. (2009) "*Heterogeneous database integration for web applications*", International Journal on Computer Science and Engineering, Vol. 1 (3), pp. 227-234.

[12] Kolaitis P. G. (2005) "Schema mappings, data exchange, and metadata management", In PODS, pp. 61–75.

[13] Bin L., Xin Z., Zhongliang D. (2011) "Database Conversion Based on Relationship Schema Mapping", Internet Technology and Applications (iTAP), 16-18 Aug. 2011, pp.1-5.

[14] Pottinger R. A., Bernstein P.A. (2002) "*Creating a mediated schema based on initial correspondences*", IEEE Data Engineering Bulletin. Vol. 25 (3), pp. 26–31.

[15] Pottinger R.A. (2004) "*Processing Queries and Merging Schemas in Support of Data Integration*", Ph.D. thesis, University of Washington, Washington, USA.

[16] Saleem, K., Bellahsene, Z., Hunt, E. (2008) "*PORSCHE: Performance ORiented SCHEma mediation*" Information System Journal, Vol. 33 (7–8), 637–657.

[17] Zaki, M.J. (2005) "*Efficiently mining frequent trees in a forest: algorithms and applications*", IEEE Trans, Knowledge Data Engineering (TKDE), Vol. 17 (8), pp. 1021–1035.

[18] Do H., Rahm E. (2002) "COMA: A system for flexible combination of schema matching approaches", International Conference on Very Large Data Bases (VLDB), Hong Kong, China, August 20–23, 2002, pp. 610–621.

[19] Do H., Rahm, E. (2007) "*Matching large schemas: approaches and evaluation*". Information System Journal, Vol. 32 (6), pp. 857–885.

[20] Rahm E., Do H, Massmann S. (2004) "*Matching large XML schemas*", ACM SIGMOD Record, Vol. 33 (4), pp. 26–31.

[21] Sarma A.D., Dong X., Halevy A. (2004) "Bootstrapping pay-as-you-go data integration systems", ACM SIGMOD International Conference on Management of Data, Vancouver, BC, Canada, 10–12 June 2008, pp. 861–874.

[22] Tansalarak N., Claypool K.T. (2007) "*QMatch: using paths to match XML schemas*", Knowledge Data Engineering, Vol. 60 (2), pp. 260–282.

[23] Nguyen H. Q., David Taniar, J. Wenny Rahayu, Kinh Nguyen (2011) "*Double-layered schema integration of heterogeneous XML sources*", Journal of Systems and Software, Vol. 84 (1), pp. 63-76.

[24] Lee Y., Sayyadian M., Doan, A. Rosenthal A. (2007) "*eTuner: tuning schema matching software using synthetic scenarios*", The International Journal on Very Large Data Bases, Vol. 16 (1), pp. 97–122.

[25] Melnik S., Rahm E., Bernstein P.A. (2003) "Rondo: a programming platform for generic model management", ACM SIGMOD International Conference on Management of Data, San Diego, California, USA, June 9-12, 2003, pp. 193–204.

[26] OAGi (2010) "Open Applications Group–OAGIS 9.4.1.", URL: http://www.oagi.org/dnn2/DownloadsandResources.aspx.

[27] Nguyen H. Q. (2008) "Repository for XML Schema Integration", La Trobe University, Australia. URL: http://homepage.cs.latrobe.edu.au/h20nguyen/research.