2010 8th IEEE International Conference on
Control and Automation
Xiamen, China, June 9-11, 2010

A Novel Approach of Computing XML Similarity based on Weighted XML Data Model

Li Shuqing, Xu Meifeng

Abstract—**In this paper we introduce a weighted XML data model based on traditional XML data model. We can use this model with the same DTD structure to express many common applications such as personalized ontology. Subsequently this paper discusses the weight value assignment methods in two situations which include comparison of XML data instants based on same DTD and expression of weighted XML data model in personalized ontology. And we design a new similarity algorithm of this weighted XML data model. At last, some experimental implementation and results are discussed.**

I. Introduction

ALONG with lots of applications and rapid development in recent years, personalized weighted ontology becomes the main focus of researchers. Though personalized weighted ontology can express users' interest model efficiently and precisely, correlative methods of design and implementation needs further research. Firstly, this paper introduces the concept and characteristics of weighted XML data model, and designs a method building personalized weighted ontology based on weighted XML data model with homo-structure-hetero-value evaluation strategy. This paper also explains the methods of constructing this model from three points, such as value design, hierarchy design and weight design of which strategy and principle is explained in detail. Finally the similarity arithmetic of this personalized weighted ontology is introduced and some test experiments are discussed for comparison.

With the development of Web applications, data transmission between different applications becomes more important and necessary. Many heterogeneous data need to be transferred for the purpose of communications [1]. We can notice that XML data, which is actually standard data format for data transmission, receive more researchers' attentions increasingly. We use XML to represent data models and construct lots of efficient applications. XML has been accepted as a major means for efficient data exchange. These correlative applications include data exchange, XML clustering, schema or ontology integration, heterogeneous data integration, personalized content delivery, message mapping, web service discovery and composition, agent communication [2].

For example, the usage of XML is the basis of semantic Web network. We utilize its internal hierarchy and semantic information of nodes to express all kinds of complicated semantic concepts and the relations between them [3]. And we can use this XML model to express the uses' interesting model for personalized recommendation technology.

Although the goals of these applications are different and the complexion of implementation is not consistent, there are some basic methods which are widely used in all applications. The similarity of XML data model is a crucial problem of them. Many researchers have given a great number of algorithms. But lots of problems still exist, especially problems about complexity and efficiency.

In our research of personalized products recommendation technology, we design a new algorithm of computing similarity of XML data model, which uses weighted XML data model. Through our experiments, we find this method has better efficiency and effectiveness. This paper will discuss this process in detail. The remainder of this paper is organized as follows. Section 2 briefly reviews some background in both XML similarity principles and methods. In section 3 we give a brief introduction of weighted XML data model. Section 4 presents the weight value assignment of weighted XML data model. Section 5 introduces the similarity algorithm of weighted XML data model in detail, followed by the experimental conclusion in Section 6, which including a description of the experimental environment and an interpretation of the results. Related works and some future research directions will be covered in Section 7.

II. Backgrounds

In formal definition, we believe that similarity is an increasing function of commonality and decreasing function of difference. Common similarity algorithms include two main types. One is lexicon-based algorithm and the other is structure-based algorithm. And many authors put forward some more advanced algorithms that we will discuss below.

Lexicon-based algorithm is more common and earlier. We can measure the similarity between different nodes based on nodes' contents [4]. Earlier researches often use the lexical information of nodes' contents with some methods such as n-gram measures and edit distance measures1. Some new

Manuscript received September 26, 2009.

Li Shuqing is with School of Information Engineering, Nanjing University of Finance & Economics, Nanjing, 210046, China (corresponding author to provide phone: 13912964688; e-mail: leeshuqing@163.com).

Xu Meifeng is with School of Information Engineering, Nanjing University of Finance & Economics, Nanjing, 210046, China (e-mail: xu_mf1990@tom.com).

researches begin to combine these methods with semantic analysis. And many researchers want to analyze the semantic information of lexicon with thesaurus and information contents to get more effective results [5].

Although structure-based algorithms need nodes' contents analysis, the key features lie in the analysis of nodes' relation and attributes' information [4]. These methods are easier to understand since we often see XML data structure as a tree and we have had some effective accessing methods such as path matching and tree edit distance (TED) [6]. Flesca et al. quantifies the structures of XML documents with time series approaches. Through the comparison with the Discrete Fourier Transformation (DFT), he can analyze these time series and get the final similarity measure at last [7]. Some other researchers use simpler methods based on path shingles. This method costs more although it is more efficient [8]. We notice that many researchers introduce many weighted methods such as weighted tag similarity measures and weighted tree similarity measures which have better experimental effects [9]. Anna Formica designs a method based on weighted type hierarchy and proposes an element similarity method which consists in the association of weights with the types of the hierarchy, standing for the probabilities that randomly selected instances are of that types [10]. To tell the truth, these methods give us some beneficial illumination.

Other methods often use some new measures to get similarity of XML. For example, some researchers put forward a new approach based on link analysis of nodes in XML structure [11]. Malet Streetif even designs a technique for measuring the structural similarity of semi-structured documents based on entropy which is the first true linear-time approach for evaluating structural similarity [12].

Of course any method has own limitations and disadvantages. For example, if two XML data instants with completely identical nodes may have very different structure. So combination of many different methods is better solution in many occasions [13].

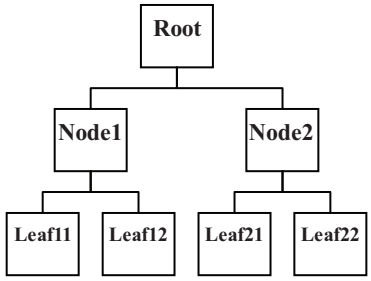
III. WEIGHTED XML DATA MODEL

Traditional XML data model always has a hierarchy structure like a tree view. In some applications such as personalized ontology, each node in XML data model can denote a user interest concept. The combination of many interesting concepts can also denote a specific personalized uses' interest mode.

What does weighted XML data model refer to? In facts, weighted XML data model is constructed based on the standard XML data model. That is to say, we can add a different weight value to each node of XML data model and all these weight values can represent the importance of corresponding nodes in any XML data instant. For conveniences, the structure of XML data model we will explain below is shown in Figure 1.

```
<Root weight="0">
<Node1 weight="0">
   <Leaf11 weight="0"></Leaf11>
   <Leaf12 weight="0"></Leaf12>
</Node1>
<Node2 weight="0">
   <Leaf21 weight="0"></Leaf21>
   <Leaf22 weight="0"></Leaf22>
</Node2>
</Root>
```

(a) XML data view



(b) Tree view

Fig. 1. The Structure of XML Data Model

The corresponding DTD structure is shown in Figure 2.

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT Leaf11 EMPTY>
<!ATTLIST Leaf11 weight CDATA #REQUIRED>
<!ELEMENT Leaf12 EMPTY>
<!ATTLIST Leaf12 weight CDATA #REQUIRED>
<!ELEMENT Leaf21 EMPTY>
<!ATTLIST Leaf21 weight CDATA #REQUIRED>
<!ELEMENT Leaf22 EMPTY>
<!ATTLIST Leaf22 weight CDATA #REQUIRED>
<!ELEMENT Node1 (Leaf11, Leaf12)>
<!ATTLIST Node1 weight CDATA #REQUIRED>
<!ELEMENT Node2 (Leaf21, Leaf22)>
<!ATTLIST Node2 weight CDATA #REQUIRED>
<!ELEMENT Root (Node1, Node2)>
<!ATTLIST Root weight CDATA #REQUIRED>
```

Fig. 2 DTD Structure of this XML Data Model

Unlike the standard XML data model, weighted XML data model can express the different importance of different nodes apart from hierarchical structure and nodes' contents. This model can have two functions. One is that it can be beneficial to compute similarity of XML data instants based on same DTD. Other is that it can help to express personalized ontology and implement recommendation services.

IV. The Weight Value Assignment of Weighted XML Data Model

A. The Weight Value Assignment of Weighted XML Data Model based on Same DTD

Some researches have been taken to measure the similarity between XML data and DTD. For example, Joe Tekli et al. propose a new approach based on the concept of tree edit distance, as an effective and efficient means for comparing tree structures, XML documents and DTD being modeled as ordered labeled trees [14]

In many applications, XML data instants often have an identical DTD structure. All XML data instants' structures conform to DTD even if some data instants are a little different. For example, according to the DTD above-mentioned, we can get two XML data instants shown in Figure 3.

```
<Root weight="3">
<Node1 weight="1">
   <Leaf11 weight="1"></Leaf11>
</Node1>
<Node2 weight="2">
   <Leaf21 weight="1"></Leaf21>
   <Leaf22 weight="1"></Leaf22>
</Node2>
</Root>
```

(a) Instant 1

```
<Root weight="2">
<Node1 weight="1">
   <Leaf12 weight="1"></Leaf12>
</Node1>
<Node2 weight="1">
   <Leaf22 weight="1"></Leaf22>
</Node2>
</Root>
```

(b) Instant 2

Fig. 3 Two XML Data Instants based on Same DTD

In these two XML data instants, we can see their structures adaptive to the same DTD. However, we should notice another important thing. Each node in these two XML data instants has an inconsistent weight value. The assigning method of weight value has these features.

1) All leaf nodes have the weight value equal 1. These values can indicate these nodes' existing in XML data instants. And we can conclude that all nodes that do not appear in data instants will have a weight value equal 0. In order to save storage, we do not add those nodes in final XML data structures.

2) All non-leaf nodes will have the weight value which is the sum of weight values of all sub-nodes. So the level of nodes is higher, the weight value of nodes is greater. And the weight value of root node can be greatest.

Besides the weight value has the ability of expressing existence of nodes, we can notice that the difference of structure can be embodied by the difference of the weight values of node. In this way, we can change traditional methods of computing similarity of XML data instants and get a new approach which does not concern the structure of XML data instant. We only use the weight value to denote the structure difference and compute similarity. Obviously, this method is easier and more manipulable than traditional one.

B. The Weight Value Assignment of Weighted XML Data Model in Personalized Ontology

In some technologies such as personalized recommendation, the key issue is how to construct a better user interest model more efficiently and more accurately. The researches often use keyword-based methods traditionally. With the development of ontology technology, researches begin to use ontology to denote user interest model. And common methods often use standard XML data model to denote personalized ontology. Because of complexion of computation, many methods often have problems such as lower performances and time-consuming problems.

Now we have a new method to denote personalized ontology with weighted XML data model. In such occasion, we only need assign a different weight value to each node in one XML data instant which denotes these personalized characteristics. For example, a personalized ontology in personalized products recommendation system is shown in Figure 4.

```
<Player ref="battery" weight="1">
<mp3 weight="0"></mp3>
<mp4 weight="2">
   <Sony weight="2"></Sony>
   <Patriot weight="2">
      <E5808 weight="4"></E5808>
      <F820 weight="0"></F820>
   </Patriot>
</mp4>
<Recorder weight="0"></Recorder>
</Player>
```

Fig. 4. A Personalized Ontology based on Weighted XML Data Model

According to the example above-mentioned, we can get two personalized ontologies shown in Figure 5.

```
<Root weight="8">
<Node1 weight="2">
   <Leaf11 weight="2"></Leaf11>
</Node1>
<Node2 weight="6">
   <Leaf21 weight="4"></Leaf21>
   <Leaf22 weight="2"></Leaf22>
</Node2>
</Root>
```

(a) Instant 1

```
<Root weight="10">
<Node1 weight="2">
   <Leaf12 weight="2"></Leaf12>
</Node1>
<Node2 weight="8">
   <Leaf21 weight="4"></Leaf21>
   <Leaf22 weight="4"></Leaf22>
</Node2>
</Root>
```

(b) Instant 2

Fig. 5 Two Personalized Ontologies based on Weighted XML Data Model

From these examples, we can see the two corresponding users have different interesting characteristics but also have some similarity in some fields. All these features can be indicated by the different weight values. And we can use these weight XML data instants to compute their similarity for implementing personalized recommendation.

V. The Similarity Algorithm of Weighted XML Data Model

Before explaining the detailed algorithm, we should introduce some basic designing principles.

1) Although weighted XML data model have three measures such as hierarchical structure, nodes' contents and weight values of nodes, we should only concern the last one, that is to say, weighted value of nodes. Because all weighted XML data instants based on same DTD have the identical structure and nodes' contents, the differences between them mainly lie in weighted values. Even if we delete all nodes having weight value with 0 and get weighted XML data instants with the different structure, we also believe that the similarity of structure can be embodied by the weighted value. As mentioned-above in two instants in Figure 5, the difference of nodes in lower levels can be embodied by their different weight values, and the similarity of nodes in higher levels can be embodied by their identical weight values.

2) We believe that the similarity between nodes in lower levels should be more important than the similarity between nodes in higher levels. In some applications such as personalized recommendation, we often denote detailed users' interests through the nodes in lowers levels. For example, we can use the amount of uses' accessing this product as its weight value. Meantime we always use nodes in higher level to denote generalized users' interests. So in similarity computation, we think it is important to embody this influence of nodes' level to similarity.

According to these principles, we design a new similarity algorithm of weighted XML data model. Assume the DTD is X, corresponding XML data instants are x1 and x2. The similarity algorithm of these two XML data instants is shown in pseudo-code below.

```
Input: X, x1, x2
Output: the similarity of x1and x2
//Normalize the weight values of all nodes
normalization(x1);
normalization (x2);

//Decay factor of level influence
decayFactor=1;

//The greatest depth in X
similarityValueInLevel[n];

//Handle each level in reversed order
for each level li of X in reversed order {
//Temporary variable for computing the similarity in level i
totalValueInLevel=0;

//Handle each node in current level
for each node nj in li {
//Smooth factor which is the total number of nodes in level i
totalNodeNumber=getTotalNodeNumberInLevel(i)

//Get the similarity of node j in level i between two XML data instants
totalValueInLevel += |njx1-njx2|/max(njx1, njx2)/totalNodeNumber
}

//Get the similarity of level i
similarityValueInLevel[i]= decayFactor * (1-totalValueInLevel);

//Higher level is, more decay degree is
decrease(decayFactor);
}

//Return the final similarity of two weighted XML data instants
return max(similarityValueInLevel);
```

VI. EXPERIMENTS

At first, we want to validate this approach with the simplest model. Two weighted XML model is shown in Figure 6.

```
<Root weight="1">
<Node1 weight="1">
   <Leaf11 weight="1"></Leaf11>
</Node1>
</Root>
```

(a) Instant 1

```
<Root weight="1">
<Node2 weight="1">
   <Leaf22 weight="1"></Leaf22>
</Node2>
</Root>
```

(b) Instant 2

Fig. 6 Two Completely Different Weighted XML Instants based on Same DTD

These two models are all based on same DTD above-mentioned in Figure 2. And we can notice that they are different from each other mostly. In this experience, we get a final similarity value which is equal to 0. The similarity value of each level is shown in Table 1.

TABLE I
THE SIMILARITY VALUE OF EACH LEVEL

Level	Similarity
2	0
3	0
4	0

Then we design two completely identical weighted XML data instants which are shown in Figure 7.

```
<Root weight="1">
<Node1 weight="1">
   <Leaf11 weight="1"></Leaf11>
</Node1>
</Root>
```

(a) Instant 1

```
<Root weight="1">
<Node1 weight="1">
   <Leaf11 weight="1"></Leaf11>
</Node1>
</Root>
```

(b) Instant 2

Fig. 7 Two Completely Identical Weighted XML Instants based on Same DTD

In this experiment, we get a final similarity value which is equal to 1. The similarity value of each level is shown in Table 2.

TABLE II
THE SIMILARITY VALUE OF EACH LEVEL

Level	Similarity
2	0
3	0.5
4	1

In order to verify this algorithm, we also design a prototype experiment based on the existed Web system WebShop developed on JSP platform.

All data of XML structure and nodes' contents are got from ODP in our experiment. Through our XML parser, we can get a full XML data structure with 756969 nodes and 15 levels. For convenience, we only extract all nodes in "Shopping" node. All sub-nodes are 5378 and the level number is 11. In experiment, all the XML data and structure are stored in relational database.

For example, relational table structure_shopping stores the information about all processed nodes as shown in Table 3.

TABLE III
SOME RECORDS IN TABLE STRUCTURE_SHOPPING

NodeID	Node Title	Level
451605	Top/Shopping	2
451842	Top/Shopping/Children	3
451843	Top/Shopping/Children/Baby	4
451844	Top/Shopping/Children/Baby/Albums_and_Frames	5
451845	Top/Shopping/Children/Baby/Bath_and_Body	5
451846	Top/Shopping/Children/Baby/Bibs_and_Towels	5

Relational table XMLdata stores all XML data instants. Each XML data instant corresponds to some records which have a weight value. For saving storage and fastening computation, we assume all nodes having a default weight value equal 0 and we do not store those nodes whose weight value is equal to 0. Relational table XMLdata is shown in Table 4.

TABLE IV
SOME RECORDS IN TABLE XMLDATA

Serial ID	XML Instant ID	Node ID	Weight Value
4	1001	451841	7
5	1001	452302	4
6	1001	455835	13
7	1005	452303	1
8	1005	453250	2

According to two XML data instants in Table 4, we can get their similarity value is 0.43438. The detailed computation process is shown in Table 5.

At last, we carry an evaluation of this approach. 15 testers

TABLE V
FINAL SIMILARITY OF EACH LEVEL

Level	Similarity
2	0
3	0.43438
4	0.35625

are asked to take part in our experimental evaluation. And we prepare 15 XML data instants. And we partition 6 levels to denote computation's effect which range from 0 to 5. We specify 0 as inaccurate completely and 5 as accurate completely. Through comparison of each two instants, testers are asked to evaluate the precision of similarity value based on our algorithm. Each tester is asked to find 5 most effective similarity values as final results. The average evaluation's results are shown in Table 6.

TABLE VI
FINAL SIMILARITY OF EACH LEVEL

Serial Number	Averaged Precision
1	3.736
2	4.005
3	3.989
4	4.041
5	3.664

VII. CONCLUSIONS AND FUTURE WORK

This paper gives a new approach for computing XML similarity based on weighted XML data model. Through the prototype experiments, we believe that it is a better method for this aim. Of course, we also find that this approach still need to be farther consummated. For example, we should design more similarity algorithm to validate and choose the best implementation method, especially in some situations which need to handle greater data collection. Moreover, we think current XML similarity is applicable for weighted XML data instants based on same DTD. For those weighted XML data instants based on different DTD, we need more research to design some more effective methods to handle them. And we believe these advanced function can be applied to more applications. So we will develop corresponding approaches in follow researches.

REFERENCES

[1] Shvaiko, P., & Euzenat, J. A survey of scham-based matching. Journal of Data Semantics IV, 3730, 2005, 14–171.

[2] Li, J., et al. Computing structural similarity of source XML schemas against domain XML schema. Australian Computer Society, Inc. Darlinghurst, Australia, Australia, 2008.

[3] J. T. Pollock, R. Hodgson. Adaptive information: Improving business through semantic interoperability. Grid Computing, and Enterprise Integration, (Wiley Series in Systems Engineering and Management), Wiley-Interscience, 2004.

[4] Jeong, B., Kulvatunyou, B., Ivezic, N., Cho, H., & Jones, A. Enhance reuse of standard e-business XML schema documents. In Proceedings of international workshop on contexts and ontology: theory, practice and application (C& O'05) in the 20th national conference on artificial intelligence (AAAI'05), 2005.

[5] Pedersen, T., Patwardhan, S., & Michelizzi, J. WordNet: Similarity-measuring the relatedness of concepts. In Proceedings of the 19th national conference on artificial intelligence (AAAI'04), 2004.

[6] Buttler, D. A short survey of document structure similarity algorithms. In Proceedings of the 5th international conference on internet computing (IC' 04), 2004.

[7] S. Flesca, G. Manco, E. Masciari, L. Pontieri, and A. Pugliese. Fast detection of XML structural similarity. IEEE Transactions on Knowledge and Data Engineering, 17(2):160–175, February 2005.

[8] D. Buttler. A short survey of document structure similarity algorithms. In 5th Int. Conf. on Internet Computing, Las Vegas, Nevada, 2004.

[9] Bhavsar, V., Boley, H., & Yang, L. A weighted-tree similarity algorithm for multi-agent systems in e-business environments. In Proceedings of the business agents and the semantic web (BASeWEB) workshop, 2003.

[10] Formica, A. Similarity of xml-schema elements: A structural and information content approach. Computer Journal, 2007.

[11] G. Guerrini, M. Mesiti, I. Sanz. An overview of similarity measures for clustering XML documents. In: A. Vakali, G. Pallis (Eds.), Web Data Management Practices: Emerging Techniques and Technologies, IDEA Group Publishing, 2006, 56–78.

[12] S. Helmer. Measuring the structural similarity of semi-structured documents using entropy. VLDB 2007.

[13] Buhwan Jeong, Daewon Lee, Hyunnbo Cho, and Jaewook Lee. A novel method for measuring semantic similarity for XML schema matching. Expert Systems with Applications, 34(3), 2008, 1651—1658.

[14] J. Tekli, R. Chbeir and K. Yetongnon. Structural Similarity Evaluation between XML Documents and DTDs. In Proceedings of the 8th International Conference on Web Information Systems Engineering (WISE'07), Springer-Verlag Berlin Heidelberg (LNCS 4831), Nancy, France, 2007, 196-201.