Belief Integration Approach of Uncertain XML Documents

Asma Hamissi
LARODEC
Université de Tunis,
ISG Tunis, Tunisia
hamissi.asma@gmail.com

Boutheina Ben Yaghlane
LARODEC
Université de Tunis,
IHEC Carthage, Tunisia
boutheina.yaghlane@ihec.rnu.tn

Abstract

Data can be available through several sources. The integration of these sources offers an easy access and manipulation of scattered data. The fast emerging of XML as a standard for representing and exchanging static and dynamic information has increased the interest of integrating XML data. Nevertheless, conflicting sources induce to conflicting values that result on uncertainty. So, how we can apply the *Dempster-Shafer theory of evidence* [9] to model and integrate XML data containing such kind of conflicts to have a belief representation and integration of uncertain XML data.

Keywords: XML data, value conflicts, uncertainty, integration, evidence theory.

1 Introduction

The uncertain nature of the data and the process of integration itself make the integration a more interesting problem but also a more difficult one. When an integration system is faced to a conflict in values, how it can react?

In our approach, we assume that instead of choosing only one alternative we can present all possible alternatives where we maintain the uncertainty of the data. Therefore, this solution will offer a more reliable and trusted integrated source.

The *probability theory* is the most used mechanism to represent and integrate uncertain XML data. Although, the *Dempster-Shafer theory of evidence* [9] allows to combine easily information from different sources, there is not a much research effort offering comprehensive XML structures holding uncertain information with it. The main effort is reported in [5], where it has been used a logical framework and fusion rules to merge uncertain XML documents.

Therefore, in this paper we propose a new approach that uses the benefits of the *evidence theory* to represent and integrate uncertain XML documents. Our proposition is different from [5] by the adopted framework and reasoning.

The major reasons that have supported us to pick out the evidence theory are the fact that this theory can be considered as a generalization of probability theory, it models easily the partial and total ignorance and the Dempster's combination rule [9] allows to combine multiple pieces of evidence. So, it facilitates the merging of uncertain XML data to which are assigned mass values.

The outline of the paper is as follows: in Section 2, we present the sources of uncertainty in integration. In Section 3, we give an overview of the evidence theory concepts that are related to our proposal. In Section 4, we propose some basics for the representation of uncertain XML data. In Section 5, we discuss

L. Magdalena, M. Ojeda-Aciego, J.L. Verdegay (eds): Proceedings of IPMU'08, pp. 370–377
Torremolinos (Málaga), June 22–27, 2008

every time one aspect of our proposition concerning the uncertain XML documents integration and we propose the corresponding algorithm. In Section 6, we illustrate our proposition with an example from the real world. Finally, we present in Section 7 our conclusion and future work.

2 What are sources of uncertainty in XML data integration ?

When we have to integrate two sources, several situations can result in uncertainty. Generally, sources of uncertainty depend on the data itself, or on conflicts that can arise at integration time.

2.1 Uncertain data

Real life data is frequently uncertain and uncertainty in XML data is due to many factors. We cite for example the following ones:

- **Information retrieval**. Unlike traditional information retrieval that was based on retrieving documents with specific keywords, in automated information extraction from unstructured or semi-structured sources, we can't avoid uncertainty which is expressed by a confidence score that indicates the system's confidence towards extracted data [6].

- **Unreliable sources**. XML data to be integrated can come from unreliable sources of information or not up to date [2].

- **Noisy sources**. XML data can be constructed from noisy input source like sensor readings, image processing and bioinformatics [4].

- **Summarized and evaluative information**. XML documents can be exploited to describe information in one or more scientific data sources. Generally, such XML documents contain summarized and evaluative and are constructed by information extraction systems [6].

2.2 Conflicting sources

When two XML documents are integrated, data in itself can be certain. However, many conflicts can arise at integration time and therefore, result in uncertainty. Two main conflict aspects are distinguished:

- We are uncertain if two elements refer to the same real world or not. This aspect is well illustrated with the case of two bibliographic references containing the title and the name of the authors of papers (see Table 1).

Table 1: Examples of bibliographic references.

BIBLIOGRAPHIC REFERENCE 1	
Title	**Author**
Data Integration	Albert H.Smith

BIBLIOGRAPHIC REFERENCE 2	
Title	**Author**
Data Integration	Albert Smith

  When these two bibliographic references are integrated there is a conflict due to the fact that we are uncertain whether the two papers refer to the same real world or not, i.e. we speak about two different papers or only one. This uncertainty is caused by the different names of the authors.

- Even if the two elements are decided to refer to the same real world, the two sources can conflict on some values. If we consider the bibliographic reference example developed above, with the condition that the two papers are considered to refer to the same real world, a conflict is detected due to the contradictory values "Albert H.Smith" and "Albert Smith" of the author element. In our work we are interested on representing and integrating XML documents containing this kind of conflicts.

3 Evidence theory concepts

The basics of the theory of evidence [9] that we have used in our work are the following:

Definition 1 ***Belief mass function****. Let Θ be a set of mutually exclusive and exhaustive hypothesis about one problem domain. Ω is called frame of discernment. The belief mass function (bmf) denoted m(A) given to the subset A express the total unitary amount of belief that supports that the actual world is in A, and does not support any more specific subset of Θ because of a lack of information. The belief mass function (bmf) can be defined as:*

$$m : 2^{\Theta} \rightarrow [0,1] \text{ such that :} \begin{cases} m(\emptyset) = 0 \\ \sum_{A \subseteq \Theta} m(A) = 1. \end{cases}$$

Remarks.

1. If m(A)>0 then the subset A is called focal element.
2. When all focal elements are singletons a mass function can be considered as a probability distribution. Therefore, evidence theory can be considered as a generalization of probability theory.
3. The total ignorance is expressed under these conditions: (i) $m(\Theta) = 1$, (ii) $\forall A \subseteq \Theta$ such that $A \neq \emptyset$, m(A)=0.
4. The partial ignorance is expressed under these conditions: (i) $m(\Theta) \neq 0$, (ii) there exists at least a subset $A \neq \emptyset$ such that $m(A) \neq 0$.

Definition 2 ***Dempster's rule of combination****. Let be m1 and m2 two mass functions from two independent sources and on the same frame of discernment. These two masses can be combined using Dempster's rule of combination that can be defined as:*

$$(m1 \oplus m2)(C) = \frac{\sum_{Ai \bigcap Bj=C}^{i,j} (m1(Ai) \times m2(Bj))}{1 - \sum_{Ai \bigcap Bj=\emptyset}^{i,j} (m1(Ai) \times m2(Bj))}$$

Remarks.

1. Let $k = \sum_{Ai \bigcap Bj=\emptyset}^{i,j} (m1(Ai) \times m2(Bj))$ the degree of conflict between evidence sources. The normalization factor is denoted by $\frac{1}{1-k}$ and it allows to avoid non zero mass from being assigned to the empty set after combination.
2. More the degree of conflict k is larger, more sources of information are in conflict. If $k = 1$, sources are completely conflicting.

4 Belief representation of uncertain XML data

When we deal with uncertainty, an uncertain XML document will be represented with a *belief tree* in which we have added some special nodes. The basics of our proposition concerning the representation of uncertain XML data are as follow:

Definition 3 ***Belief XML subtree (BST).*** *A belief subtree is a tree in which are added specific kinds of nodes to express uncertainty. The root node of a belief subtree n is named "possibilities". From this node are rooted child subtrees (ST') that represent all possibilities for the value of one node. Each child subtree is rooted by a "possibility" node n'. Each possibility node has a "mass" attribute that expresses the amount of belief supporting the possible value of the node. A belief subtree can be formalized as follows:*
Let N be a set of nodes.

$$\begin{cases} BST = \{n, ST, mass\} \text{ such that:} \\ n \in N \text{ and type(n)="possibilities"} \\ \text{Each } ST = (n', ST') \text{ such that:} \\ n' \in N \text{ and type(n') = "possibility"} \\ \sum_{ST \subseteq BST} m(ST) = 1. \end{cases}$$

Remarks.

1. We mean by "possibility" an alternative resulted from two conflicting values and not the term "possibility" used in the *possibility theory* [3].

2. The possibilities must be mutually exclusive and exhaustive.

Definition 4 ***Belief XML tree (BT).*** *A belief tree must contain at least one belief XML subtree.*
Let N be a set of nodes. A belief tree can be formalized as follows :

$$BT = (n, ST) \text{ such that it exists at least one } BST \subseteq ST.$$

Definition 5 ***Certain belief XML subtree (CBST).*** *A certain belief subtree is a belief subtree in which there is only one possibility for a node value. The subtree representing this possibility is a singleton and the mass value is equal to 1.*

Definition 6 ***Certain belief XML tree (CBT).*** *A certain belief XML tree is composed only by certain belief XML subtrees (CBSTs).*

Definition 7 ***Partial ignorance representation.*** *The partial ignorance in a belief tree for the value of a node is expressed with: $m(\Theta) \neq 0$ and it must exist at least one singleton possibility where the mass value is different from 0.*

Definition 8 ***Total ignorance representation.*** *The total ignorance in a belief tree concerning the value of a node is expressed with only one BST containing all of the propositions (Θ) such that $m(\Theta)$=1.*

Example 1 *Figure 1 is an example of a belief tree representing an uncertain XML document composed by only one belief subtree. This example models also the case of the partial ignorance.*

5 Belief integration of uncertain XML data

The belief integration of two uncertain XML documents algorithm denoted *integrate-BT* requires as input two uncertain XML documents (*X-doc1* and *X-doc2*) that contain information about an element from the same

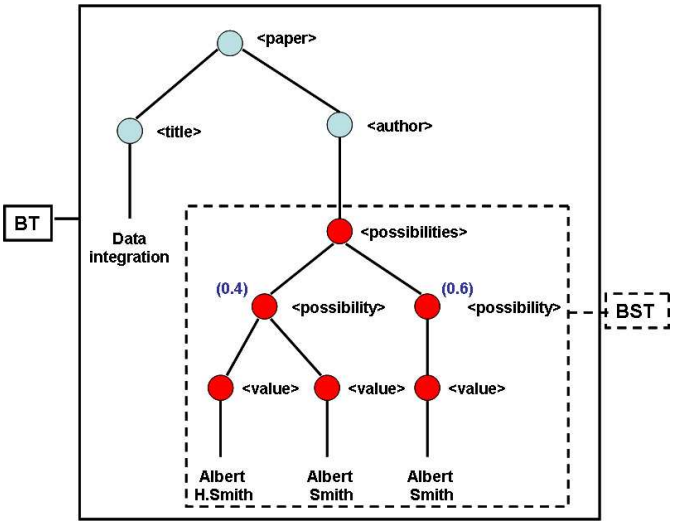


Figure 1: Example of a belief tree

real world object and it provides as output an uncertain integrated XML document (*result-doc*). As a first step, the *integrate-BT* algorithm transforms these two documents to belief trees (BT_1, BT_2) by using the *input-file* function (Figure 1 is an example of a belief tree). Then, it extracts every pair of belief subtrees (BST_i, BST_j) belonging to the two input belief trees. If there is not uncertainty about the value of the corresponding element, then this subtree is copied on the resulted integrated tree (IBT). Else, the principal algorithm applies the *integrate-BST* algorithm. It requires as input two BSTs from two different sources. It extracts from each subtree subsets of values and assigned masses, then it combines them and produces a merged belief subtree ($IBST_{ij}$) containing combined subsets and masses by using the *combination* procedure. At the last step of the principle algorithm, the IBT is saved in an XML document (*result-doc*) with the *save* function.

5.1 Basic functions

We subsume the existence of some basic functions that manage a node properties: level, type, name, value and length.

- The *level function* returns the position of a node within an XML tree. The level must be an integer between 1 and the number of nodes of the tree.

- The *type function* returns the kind of a node either *element* or *leaf*.
- The *name function* returns a string representing the content of an element node.
- The *value function* returns a string representing the content of a leaf node.
- The *length function* returns the number of nodes of an XML tree.

5.2 integrate-BT algorithm

The algorithm that allows to integrate two uncertain XML documents is as follows:

Algorithm 1 integrate-BT

```
BT1⟵input-file(X-doc1)
BT2⟵input-file(X-doc2)
create empty tree (IBT)
len1⟵length(BT1)
len2⟵length(BT2)
set-root(root(BT1),IBT)
i⟵2
j⟵2
while (i ≤ len1 and j ≤ len2) do
  BSTi⟵ extract(BT1,i)
  BSTj⟵ extract(BT2,j)
  if length(BSTi)≤2 and length(BSTj)≤2
  then
    IBT⟵copy-segment(BSTi,IBT,1,1)
  else
    IBSTij⟵integrate-BST(BSTi,BSTj)
    IBT⟵copy-segment(IBSTij,IBT,1,1)
  end if
  i⟵i+length(BSTi)
  j⟵j+length(BSTj)
end while
result-doc ⟵ save(IBT,file-name)
```

The algorithm is based on some functions and procedures that are the following:

- *Extract procedure* returns a subtree from a tree given the level of its root. So, it requires as input the tree and the level from which it will extract the subtree.
- *Copy-segment function* allows to extract a subtree from a first tree given the level of its root then, it will be inserted in a specified position (level) in a second tree.

5.3 integrate-BST algorithm

The algorithm that allows to integrate a pair of uncertain XML subtrees is the following:

Algorithm 2 integrate-BST

```
len1⟵length(BST1)
len2⟵length(BST2)
create empty tree (IBST)
set-root(root(BST1),IBST)
i⟵1
j⟵1
while (i≤len1 or j≤len2) do
  if type(nodei) is element then
    if name(nodei) is "possibilities" then
      BST1⟵extract(BST1,i)
      BST2⟵extract(BST2,j)
      BSTij⟵combination(BST1,BST2)
      copy-segment(BSTij,IBST,length(IBST))
      i⟵i+len1
      j⟵j+len2
    else
      add nodei to IBST
      i⟵i+1
      j⟵j+1
    end if
  else
    add nodei to IBST
    i⟵i+1
    j⟵j+1
  end if
end while
```

The *combination procedure* uses the *Dempster's combination rule* to merge multiple mass values from different sources. It requires as input two BSTs from two different sources. It extracts from each subtree subsets of values and assigned masses, then it combines them and produces a merged BST containing combined subsets and masses.

6 Illustrative example

The web XML documents are one of the examples that illustrate the conflicting nature of the data. In fact, if we consider the integra-

tion of bibliographic sources (see Section 2.2) represented in the form of XML documents we are usually faced to mismatches and variant spellings.
To more illustrate this idea, let us look at the example in figures 2 and 3 and suppose that we have to integrate these two documents by using the *integrate-BT algorithm*. The two documents represent uncertain information due to the conflicting values of the authors of the paper.

- **Source 1**. The first source contains two possibilities. The subsets and assigned mass values are the following:
  $m_1\{AlbertH.Smith\}$=0.8
  $m_1\{AlbertSmith\}$=0.2
- **Source 2**. The second source contains also two possibilities illustrating the case of partial ignorance. The subsets and assigned mass values are the following:
  $m_2\{AlbertH.Smith, AlbertSmith\}$=0.4
  $m_2\{AlbertSmith\}$=0.6

```
<paper>
 <title>Data integration</title>
 <author>
  <possibilities>
   <possibility mass=« 0.8 »>
     <value>
       Albert H.Smith
     </value>
   </possibility>
   <possibility mass=« 0.2 » >
     <value>
       Albert Smith
   </value>
   </possibility>
  </possibilities>
 </author>
</paper>
```

Figure 2: XML Document 1

At the first step, the *integrate-BT* algorithm transforms these two documents to belief trees like the one in Figure 1. Then, the nodes where there is not uncertainty about their values are copied in the *IBT* ($\prec$paper$\succ$ and $\prec$ title$\succ$).
When the algorithm reaches the $\prec$author$\succ$ node, it extracts the two BSTs rooted from this node and it applies the *integrate-BST* algorithm. After that, the subsets and mass

```
<paper>
 <title>Data integration</title>
 <author>
  <possibilities>
   <possibility mass=« 0.4 »>
     <value>
       Albert H.Smith
     </value>
     <value>
       Albert Smith
     </value>
   </possibility>
   <possibility mass=« 0.6 » >
     <value>
       Albert Smith
     </value>
   </possibility>
  </possibilities>
 </author>
</paper>
```

Figure 3: XML Document 2

values are extracted from each BST.
The calculation of the combined masses are detailed in Table 2. We suppose in this table that *val1* refers to the value "Albert H.Smith" and *val2* to "Albert Smith". The degree of conflict between the information sources k = $m(\emptyset)$ = 0.48 and final results are the following:

- $m\{AlbertH.Smith\}$ = 0.32/(1 - 0.48) = 0.62.
- $m\{AlbertSmith\}$ = (0.08 + 0.12)/(1 - 0.48) = 0.38.

Table 2: Calculation of combined masses.

	$m_1\{val1\}$=0.8	$m_1\{val2\}$=0.2
$m_2\{val1, val2\}$=0.4	$m\{val1\}$=0.32	$m\{val2\}$=0.08
$m_2\{val2\}$=0.6	$m(\emptyset)$=0.48	$m\{val2\}$=0.12

After the combination, the resulted belief tree (see Figure 4) contains two possibilities in which the partial ignorance is eliminated and the uncertainty is reduced (we have now more belief that the true author is Albert H.Smith than Albert Smith).

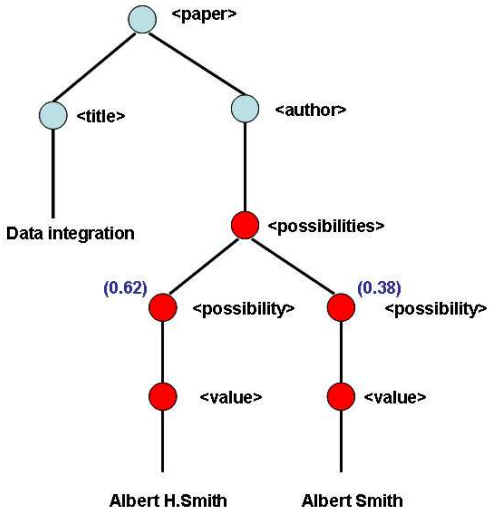


Figure 4: Resulted integrated belief tree

7 Prototype and tests

7.1 Prototype

The prototype implementation of the belief representation and integration of XML data was developed in *M*atlab 7.1.0 by using an external toolbox named XMLtree toolbox[1]. This toolbox allows to access and manipulate XML documents as a tree structure, it allows also to convert an XML tree to Matlab structure and vice versa.

7.2 Tests

The test of the algorithm was based on the following variables:

- The length of the input XML documents (number of nodes).
- The difference in the XML documents lengths.
- The degree by which the documents differ (the number of conflicts).

We tried to study the effect of the variation of each of these variables on the running time and the length of the integrated XML document by the way of three tests (see Table 3).

[1]http://www.artefact.tk/software/matlab/xml/

Table 3: Test variables.

Document length	14; 27; 53; 105; 261; 521; 1041 and 2081 nodes
Difference in documents lengths	All the combinations of the document lengths
Conflict Degree	0%; 1%; 5%; 10%; 20%; 50% and 100%

The input of the first test (see Figure 5) was two XML documents having the same length, the same depth and the same degree of conflict. But, we increased the length of these documents in every iteration.

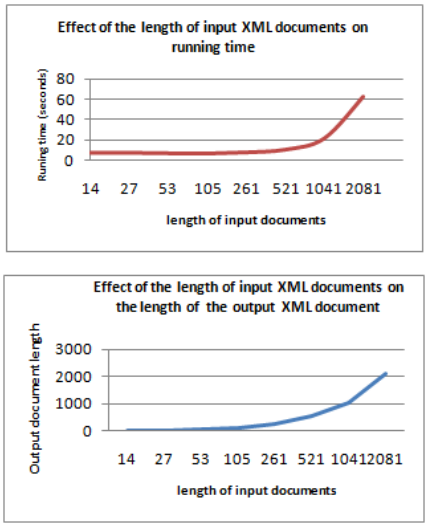


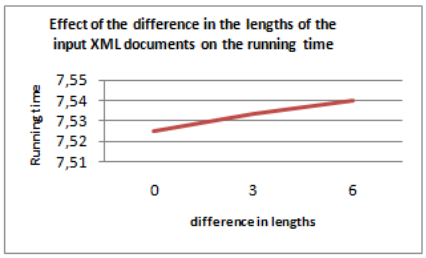
Figure 5: Test 1

The input XML documents in the second test (see Figure 6) contained the same conflicts either in the location or the degree but different sizes.

In the last test (see Figure 7), the experimentations were done on XML documents in which we increased in every iteration the number of conflicts. The length of these documents is 224 nodes in which there are 100 textentries (leaf nodes).

7.3 Interpretations

The first and the second tests have shown that there is almost no effect of the variation of the number of nodes or the lengths difference between the input XML documents on the run-



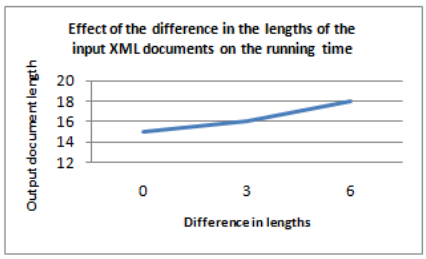
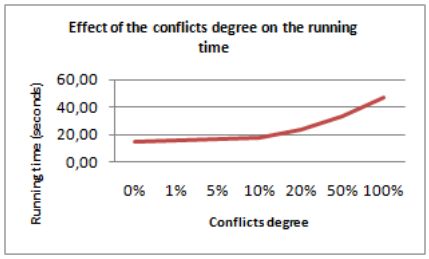


Figure 6: Test 2



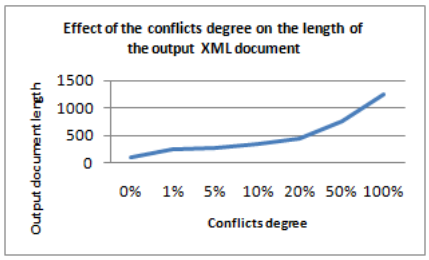


Figure 7: Test 3

ning time or the length of the resulted XML documents (see figures 5 and 6). However, the third test has demonstrated that if we increase the degree of conflict in the content of the leaf nodes between the input XML documents, then there is almost no effect on the running time. Whereas, the length of the integrated XML document will be importantly increased (see Figure 7).

8 Conclusion

The representation and integration of uncertain XML data is a recent area of research and an interesting problem. However, there is not a lot of work done dealing with this issue based on the belief functions theory. For this reason, we have tried in this paper to propose a new approach using the evidence theory to represent and integrate uncertain XML data in which uncertainty is due to conflicting values for elements representing the same real world.

As a future work, we will try to make the approach more generic by treating other problems such that the problem of determining whether two elements are referring to the same real object or not when we are faced to conflicting values.

References

[1] Bos, W. Probabilistic XML integration; the sequel (memory usage revisited). Technical report, University of Twente, Faculty of Electrical Engineering, Mathematics and Computer Science, 2007.

[2] Dong, X., Halevy, A., and Yu, C. Data Integration with Uncertainty. In *proceedings of VLDB2007*, pages 687-698, Vienna, Austria, 2007.

[3] Dubois, D. and Prade, H. Possibility theory. Plenum Press, New-York, 1988.

[4] Hung, E., Getoor, L., and Subrahmanian, V. PXML: A Probabilistic Semistructured Data Model and Algebra. In International Conference on Data Engineering, pages 467-478, Ban-galore, 2003.

[5] Hunter, A. and Liu, W. Merging Uncertain Information with Semantic Heterogeneity in XML. Knowledge and Information Systems, 9(2):230-258, 2006.

[6] Hunter, A. and Liu, W. Representing and merging uncertain information in XML: A short survey. Technical report, 2006.

[7] van Keulen, M., de Keijzer, A., and Alink, W. A Probabilistic XML Approach to Data Integration. In *International Conference on Data Engineering (ICDE)*, Tokyo, Japan, 2005.

[8] Nierman, A. and Jagadish, H. ProTDB: Probabilistic Data in XML. In *Proceedings of the 28th VLDB Conference*, Hong Kong, China, 2002.

[9] Shafer, G. A Mathematical Theory of Evidence. Princeton University Press, 1976.