Representing and merging uncertain information in XML: A short survey

Anthony Hunter[1] and Weiru Liu[2]

[1]Department of Computer Science
University College London
Gower Street, London WC1E 6BT, UK
[2]Computer Science
Queen's University Belfast
Belfast, Co Antrim BT7 1NN, UK

Abstract. XML has been used extensively on the Web for representing and exchanging a variety of static and dynamic information, such as medical records of patients, database query results, and predictive models obtained from data mining or intelligent analysis tools. Nearly every vendor of data management tools nowadays has delivered some kind of XML support which enables data in other forms to be presented in the XML format. With the increasing role that XML plays on the Web, the need to represent uncertain information has rapidly emerged too, since in real life, information is often uncertain and incomplete. In this short survey paper, we examine the state of the art of approaches to representing and managing uncertain information in XML. We look at various proposals as how uncertain information should be modelled, merged, and explore their advantages and limitations. The issue of automated generation of XML documents with uncertain information will be investigated as well.

Keywords: Semi-structured information fusion, uncertain information in XML, probabilities, belief functions

1 Introduction

Extensible Markup Language (XML) has become an important part of Semantic Web, due to its simple and flexible format. An XML document is constructed based on a DTD or an XML Schema that specifies how tags in an XML should be arranged. Initially mainly used to store and exchange static data, such as, metadata standards by Dublin core, XML is now playing an increasingly important role in the exchange of a wide variety of dynamic data too, data that are retreived or obtained upon requests. Typical examples of this kind are [11], [37], and [47], where the former constructs an XML document from a collection of multimedia data about a patient and the latter two generate XML documents that store probabilistic query results and predictive models obtained from data mining or intelligent analysis tools respectively.

To facilitate the modelling of various types of data in XML, the need to represent *uncertain* data emerged too, as in the case happened to traditional databases where numerous approaches were proposed to create and manipulate probabilistic databases (e.g.

[12, 7]), and where the OLAP technology is extended to model uncertain and imprecise data [6]. Because XML documents are structured, uncertain information associated to data must be naturally assigned, interpreted and structured. Uncertainty can occur at different levels of granularity and uncertainty can be interpreted in different ways, such as in terms of probabilities, probability intervals, reliabilities, or even beliefs. Furthermore, an integration result of XML documents having data values with certainty may create an XML document with uncertain data. Therefore, managing uncertain data in XML raises many challenging issues.

Modelling uncertainty The first challenge is how to model data that are uncertain. In some of the Web information exchange standards, uncertainties associated with part of the data (or information) are simply quoted using tag ⟨uncertainty⟩ or something similar. The TEI ([2]), **T**ext **E**ncoding **I**nitiative (TEI) initially launched in 1987, is an international and interdisciplinary standard that helps libraries, museums, publishers, and individual scholars represent all kinds of literary and linguistic texts for online research and teaching, using an encoding scheme in XML. To indicate which description or assertion or a concept is not certain, a tag with name *note* having value *uncertainty* is inserted into an appropriate place to record the uncertainty. An example of using this note tag could be

⟨persName⟩Elizabeth⟨/persName⟩ went to ⟨placeName id="p1"⟩Essex⟨/placeName⟩.
She had always liked ⟨placeName id="p2"⟩Essex⟨/placeName⟩.
⟨note type="uncertainty" resp="MSM" target="p1 p2"⟩
It is not clear here whether ⟨mentioned⟩Essex⟨/mentioned⟩ refers to the place or to the nobleman. If the latter, it should be tagged as a personal name. -MSM
⟨/note⟩

The context in ⟨note⟩ adds explanations to the part of information that is regarded not certain, e.g., Essex as a name of a place is not absolutely certain.

Yet in [13], an XML-based approach to storing and exchanging experimental and critically evaluated thermophysical and thermochemical property data with uncertainty was reported. Uncertain values are divided into different categories, such as, standard uncertainty, expanded uncertainty, level of confidence. Different tagnames are designated to indicate these different uncertain values. Similarly, [8] describes how materials property data which may be uncertain are composed into MatML, an extended XML format, A segment of MatML document encoding uncertainty on a particular type of carpet is given in Figure 1.

Obviously, all the above formats are too simple for most of the applications where uncertainty can be in heterogeneous forms and is associated with particular values with possible constrains on uncertainty distributions. Research effort offering comprehensive XML structures holding uncertain information is reported in various papers (e.g., [31, 19, 23, 25, 30, 1, 9]. A common feature among these papers is that some specific tagnames for representing uncertain information are explicitly created, and usually these tags have particular meanings when manipulating uncertainties.

Two typical examples of integrating probabilistic uncertain information into XML documents are [31] and [19]. In [31], a probability value can either be assigned to a leaf node (a textentry) or a tagname, and a single XML document can have many probability values attached to leafs and tagnames at different levels of granularity. As a result, the main focus of the research is on calculating the final probability of a value

```
⟨PropertyData property=pr4 technique=m1 source=ds1 specimen=s1 test=t1⟩
  ⟨Data format=float⟩60⟨/Data⟩
  ⟨Uncertainty⟩
    ⟨Value format=float⟩7.7⟨/Value⟩
    ⟨Units name=ug/m^2/h⟩
    ⟨Unit⟩⟨Name⟩ug⟨/Name⟩⟨/Unit⟩
    ⟨Unit power=-2⟩⟨Name⟩m⟨/Name⟩⟨/Unit⟩
    ⟨Unit power=-1⟩⟨Name⟩h⟨/Name⟩⟨/Unit⟩
    ⟨/Units⟩
  ⟨/Uncertainty⟩
    ⋮
⟨/PropertyData⟩
```

Fig. 1. An simple XML document with uncertainty values

(of a tagname) given all the probabilities available in the XML. In contrast, in [19], the main focus of the research is on integrating multiple XML documents which may contain no probabilistic information initially, but the integration result can lead to some values being uncertain. Therefore, the effort of the paper is on how to generate a merged XML document and how to obtain the posterior probabilities after merging.

Since precise probabilistic values are not easy to obtain in practice, other numerical values are often used, such as probability intervals, or mass functions in Dempster-Shafer theory of evidence (DS theory) which can be regarded as an extension of probability theory. In [30], DS theory is deployed to represent and calculate uncertain information associated with subsets of possible values of tags. Assuming that each piece of evidence provides beliefs on some subsets of possible values of a tag, multiple pieces of evidence accumulated on the same tag are combined in DS theory to take into account the effect of all the evidence. It also provides a way to calculate probability intervals of collections of values across tags.

Attempts to model uncertainty using fuzzy techniques in XML are reported in [9] and [1]. In the former, numerical values representing the importance of tags are attached to tagnames. These values are interpreted in fuzzy theory and used to calculate the importance of a set of tagnames in comparison to other sets of tagnames, so that more important information can be used first to make decisions. In the latter, XML Miner is introduced. XML Miner is a collection of tools for mining data and text expressed in XML, extracting knowledge and re-using that knowledge in products and applications in the form of fuzzy logic expert system rules. For example, in metarules and fuzzy inference engine XML rules, numerical quantities are described as fuzzytuples with the following format. In addition, for categorical outputs, the system supplies both the most likely category, with associated uncertainty, and an ordered list of alternate categories (if they exist) annotated with confidence levels.

```
⟨fuzzytuple⟩
        ⟨constant⟩0.5⟨/constant⟩
        ⟨constant⟩1.0⟨/constant⟩
```

⟨constant⟩1.5⟨/constant⟩
⟨constant⟩2.0⟨/constant⟩
⟨/fuzzytuple⟩

Data are often contributed by various sources over time. Imprecise and uncertain information is often described in different forms too, such as, probabilistic, possibilistic, fuzziness, probabilistic interval, beliefs. Therefore, developing a formal approach to handling these all different measures seems inevitable. A logic-based framework presented in [23, 24] aims at establishing a formal structure that can facilitate uncertainty reasoning in formal logics that in turn make use of knowledge in the background knowledgebase to assist querying and merging. The framework has proved to be capable of modelling a variety of forms of uncertainty and has advantages over approaches [19, 31, 30].

Looking beyond XML, under the umbrella of Semantic Web, uncertainty reasoning has attracted increasing attention in recent years. A number of research initiatives have been carried out to make the reasoning on the Web more uncertainty tolerant.

To make Web information more meaningful, a proposal was reported in [32] which integrates probabilities into DAML+OIL, a commonly used ontology language in the Semantic Web. Uncertain statements are marked with probability values instead of assuming that every statement is either true or false as in the current language format. Likewise, [22] proposed a method to model uncertainty in the Semantic Web taxonomies where concepts cannot be organized in crisp subsumption hierarchies.

Description logics (DLs) are highly regarded as the carrier of ontological knowledge on the Semantic Web. To capture uncertainties in ontological knoweledge, DLs need to be extended too, such as [41–43, 29], where fuzzy logics and probabilities are introduced into DLs for this purpose.

Merging uncertainty The second challenge is how to integrate or merge uncertain data from different XML documents. We have seem much work on modelling uncertain information in XML above, however, not much work has been done in integrating this kind of information except that were reported in [23, 19, 30]. This is a big contrast to the very active area of XML data integration without uncertainty (e.g. [14, 35, 45, 34]). One of the difficulties of merging uncertain information in XML is to preserve the properties of underlying uncertain reasoning mechanism being used, in addition to the existing challenges facing XML data integration [28, 39].

In [19], pairs of (tag, value) in an XML document and the combination of these pairs are treated as *possible worlds*. The merging of two probabilistic XML documents is to generate all the combinations of possible worlds from the two documents. As a consequence, there can be a huge number of branches in the merged XML document and there can be varieties of the document. For instance, one example given in the paper consists of two simple XML documents about `persons` with certainty (no probabilities). One document has details for four `persons` with each person has tags `firstname`, `lastname`, `phone`, `room` and associated values, the other document has details for two `persons` with the same set of tag names and corresponding values. Interestingly, merging these two simple files in [19] generates 3201 possible worlds. Most of the branches in the tree are completely meaningless.

In [23], a logic-based merging tool was proposed which uses a set of fusion rules coupled with background knowledge in the process of merging to guarantee the relevance and correctness of fusion. For instance, for the above example with information for people, if only persons with the same firstname and lastname are interpreted as possibly referring to the same person, then the merging result is a very simple XML document with four segments for four persons containing some probability components indicating multiple values for same tags, such as `room` or $tel - number$. Therefore, it is a much simpler and effective way to merge such information.

In [30] patients medical history information and current disease diagnostic information is modelled using XML documents. Since information on medical history of a patient was collected over time and patients may not remember if they have had some diseases when asked, this information can be uncertain or ignorant. Mass functions in DS theory are assigned to subtrees of an XML document as measures of the uncertainty of information being held for the patient. Each subtree (called data forest in [30]) corresponds to a subset of values of a tage, these values can be elementary values, like "blue, red" to tag "color", or compound values to a high level tag like "medical-history". Collections of subtrees therefore represent all the possible combinations of values. When assigned to disjoint subtrees, being treated as independent, mass functions are combined using a cross product operator on the set of Cartersan product of the original individual subtrees.

Querying uncertainty The third challenge is querying uncertain data in one or multiple XML documents. Obviously, if uncertainty can be associated with data values at different levels of granularity, then querying such an XML document will need to consider how to manipulate uncertainties. In [31], querying a probabilistic XML document is done under the assumption that probabilities assigned to different nodes are independent. The final probability of a value of a tag is thus the result of multiple conditional probabilities on its ancester's probabilities, until to the root tag. In [30], the degree of beliefs on a value of a tag is calculated by using the *coarsening* operator if the mass function involved is assigned to a larger subtree. In [25], a *discount* operator in DS theory is used to calculate the degree of beliefs of a value of a tag after taking into account the reliability of the initially assigned mass functions (or probabilities). All these approaches to some extend can deal with uncertain information given to different levels of granularity, when certain assumptions are met.

The rest of the paper is organized as follows. Section 2 discusses where XML documents with uncertain information may come from. Section 3 reviews the probabilistic approaches presented in [31, 19] and Section 4 introduces a mass function based method in [30]. Section 5 investigates the logic based fusion framework [23] which can both model and merge uncertain data with different uncertainty reasoning theories. It will also demonstrate how the other methods introduced in the previous sections can be subsumed in this general framework. Section 5 summarizes the paper.

2 XML with uncertain information: where are they from?

Many sources can contribute to the uncertainty of data or information being held in XML documents. In this section, we examine a number of situations where uncertainty in information is inevitable.

Information Retrieval Traditional information retrieval may be understood as retrieving documents containing specific keywords. In recent years, however, there have been major research activities in automated information extraction from text documents. The extracted information often has some form of certainty or probabilistic measures associated with it. For example, in [37], a web-based question answering system (NSIR) was implemented where the search of relevant documents is performed through the evaluation of combinations of Document + Sentence + Phrase, proximity + qtype. A probabilistic table with relevant phrases is created to rank the retrieved information, given a query. This system is used to populate a probabilistic XML database described in [31] by repeatedly asking NSIR appropriate follow-up questions.

Summaritive and evaluative information XML documents can be used to represent semi-structured information that describes information in one or more scientific datasources (such as journals, databases of empirical results, etc). Such an XML document may contain *summaritive information* about the datasource (e.g. information from an abstract, summary of techniques used, etc) plus *evaluative information* about the datasource (eg. delineation of uncertainties and errors in the information source, qualifications of the key findings, etc). These documents can be constructed by hand, by information extraction systems (e.g. [10]), or as the result of querying and analysing scientific databases in [34].

In [18], a system called *Persival* was developed which aims at providing tailored presentation of relevant medical literature for both physicians and lay consumers. Based on a user's query, the system takes documents (including images and video) as input, and generates one or more paragraphs of summary from the input documents, highlighting the common points and the differences among these input documents. The summaries can also be provided at different levels of granularity depending on who the user is. Each summary follows a fixed structure including *introduction, methods, results, and discussion*. For documents with patient medical records, the output is in a more structured format which can be easily represented with XML documents. Already in [34], the query results of medical journals are directly expressed as XML documents and these results are merged to reduce incompleteness and error messages. Another example of this kind is [11] in which temporal clinical semi-structured information is first modelled in a graphical model and then translated into XML documents.

Querying and using data in datasets The third source that contributes to XML documents containing uncertain information is querying databases, no matter the databases have uncertain data or not. For example, there are many online information resources available for bioinformatics, most of the information is in semi-structured format and may consists of uncertain information. For instance, the Cancer Genome Anatomy Project [4] tries to establish associations between Tags and Genes and between Genes and Functions. It creates tables connecting tags and genes with probabilities. Thus, a table may have the following set of attributes (Tag, Gene, ..., Probability) which shows the probabilities of associations between Tags and Genes. [17] provides tools querying

such statistical and probabilistic summaries of data, and produces probabilistic answers to queries. Either the information in the original data table or in the result of queries can be easily stored in XML format, thus creating XML documents for further uses.

Data mining is often used to develop predictive models from data, but rarely addresses how to use these models. At the same time, predictive models are often complex and ordinary users face great difficulties understanding them. In order to make the models more usable, predictive models from machine learning or data mining approaches are represented using XML documents in [47], and then used through a web-based or Palm handheld-based decision shell. For instance, if a predictive model for the diagnosis of organ-confined tumor is accessible through the decision shell, then given the data of a particular patient required by the model, a probabilistic result of diagnosis will be returned after running the model. The probabilistic result can be saved in an XML document either for immediate decisions or for future references. Effort reported in [47] can be seen as a special case of [3] which is creating a standard Predictive Modelling Makeup Language (PMML) by the Data Mining Group (DMG) based on XML for researchers in machine learning and data mining communities to store, cross-use and compare their predictive models. With more predictive models available, different aspects of tests of a single patient can be obtained which form a collection of XML documents with uncertain information (or knowledge). Taking the results from different classifiers (predictive models) as evidence, the combination of them usually deploys a technique that is capable of merging uncertainties implied in the results [5].

There are many other applications that often generate output with uncertainty. Some of these examples are image processing (e.g., [33, 36]), fault diagnosis (e.g., [38, 44]), and many more as listed in [31] and [21]. In summary, with XML being increasingly used as a standard data modelling and exchanging format for all kinds of data, and with the fact that some of these data values may be uncertain, establishing a standard structure for representing, merging, and querying uncertain information in XML documents is a pressing task in the Web era.

3 Probabilistic XML

In [31], a probabilistic XML model was presented to deal with information with uncertainty that was in the form of probabilities. Two types of probability assignments are distinguished, mutually exclusive or not mutually exclusive. For the first type, probabilities are assigned to single atoms where only one of these atoms can be true, and the total sum of probability values is less than or equal to 1 (as for ⟨`precipitation`⟩). For the second type, two single atoms can be compatible, so the total sum of probabilities can be greater than 1 (as for ⟨`cities`⟩). Using this model, we can construct an XML report as illustrated in Figure 2.

This model allows probabilities to be assigned to multiple granularities. When this occurs, the probability of an element is true is conditioned upon the existence of its parent (with probability), and so on until up to the root of the tree. A query is answered by tracing the relevant branches with the textentries specified by the query, and calculating probabilities using the conditional probabilities along these branches. These derived probabilities are then either multiplied or added depending on whether the “and” or the

```
⟨report⟩
 ⟨source⟩TV1⟨/source⟩
 ⟨date⟩19/3/02⟨/date⟩
 ⟨cities⟩
  ⟨city Prob = "0.7"⟩
      ⟨cityName⟩London⟨/cityName⟩
      ⟨precipitation⟩
       ⟨Dist type = "mutually − exclusive"⟩
        ⟨Val Prob = "0.1"⟩sunny⟨/Val⟩
        ⟨Val Prob = "0.7"⟩rain⟨/Val⟩
       ⟨/Dist⟩
      ⟨/precipitation⟩
  ⟨/city⟩
  ⟨city Prob = "0.4"⟩
      ⟨cityName⟩GreaterLondon⟨/cityName⟩
      ⟨precipitation⟩
       ⟨Dist type = "mutually − exclusive"⟩
        ⟨Val Prob = "0.2"⟩sunny⟨/Val⟩
        ⟨Val Prob = "0.6"⟩rain⟨/Val⟩
       ⟨/Dist⟩
      ⟨/precipitation⟩
  ⟨/city⟩
 ⟨/cities⟩
⟨/report⟩
```

Fig. 2. An XML report using the framework in ProTDB

"or" operation are used in the original query. For instance, the query "London is either sunny or rain on 19/3/02" is evaluated as:

$$
\begin{aligned}
&\mathtt{Prob}(\mathtt{cityName} = \mathtt{London} \wedge (\mathtt{precipitation} = \mathtt{sunny} \vee \mathtt{precipitation} = \mathtt{rain}) \\
&\qquad \wedge \mathtt{data} = 19/03/02) \\
&= \mathtt{Prob}((\mathtt{cityName} = \mathtt{London} \wedge \mathtt{precipitation} = \mathtt{sunny} \wedge \mathtt{data} = 19/03/02) \\
&\qquad \vee (\mathtt{cityName} = \mathtt{London} \wedge \mathtt{precipitation} = \mathtt{rain} \wedge \mathtt{data} = 19/03/02)) \\
&= \mathtt{Prob}(\mathtt{cityName} = \mathtt{London}) * \mathtt{Prob}(\mathtt{precipitation} = \mathtt{sunny}) \\
&\qquad * \mathtt{Prob}(\mathtt{cityName} = \mathtt{London} \wedge \mathtt{precipitation} = \mathtt{sunny} \mid \mathtt{city}) * \mathtt{Prob}(\mathtt{city} \mid \mathtt{cities}) \\
&\qquad * \mathtt{Prob}(\mathtt{cities} \mid \mathtt{report}) * \mathtt{Prob}(\mathtt{data} = 19/03/02) * \mathtt{Prob}(\mathtt{data} = 19/03/02 \mid \mathtt{report}) \\
&\qquad * \mathtt{Prob}(\mathtt{report}) \\
&\quad + \mathtt{Prob}(\mathtt{cityName} = \mathtt{London}) * \mathtt{Prob}(\mathtt{precipitation} = \mathtt{rain}) \\
&\qquad * \mathtt{Prob}(\mathtt{cityName} = \mathtt{London} \wedge \mathtt{precipitation} = \mathtt{rain} \mid \mathtt{city}) * \mathtt{Prob}(\mathtt{city} \mid \mathtt{cities}) \\
&\qquad * \mathtt{Prob}(\mathtt{cities} \mid \mathtt{report}) * \mathtt{Prob}(\mathtt{data} = 19/03/02) * \mathtt{Prob}(\mathtt{data} = 19/03/02 \mid \mathtt{report}) \\
&\qquad * \mathtt{Prob}(\mathtt{report}) \\
&= 1.0 * 0.1 * 0.7 * 1.0 * 1.0 * 1.0 * 1.0 * 1.0 + 1.0 * 0.7 * 0.7 * 1.0 * 1.0 * 1.0 * 1.0 * 1.0 \\
&= 0.07 + 0.49 = 0.56.
\end{aligned}
$$

The main advantage of this model is that it allows probabilities to be assigned to multiple levels of subtrees and provides a means to calculate the joint probability from them. However, it does not consider merging multiple probabilistic XML documents on the same issue.

Another method to model and reason with probabilistic XML information is reported in [19]. In this paper, three types of tags are identified as: (1) tags that stand for probabilities (denoted as ∇); (2) tags that stand for possible values associated with

probabilities (denoted as ○); and (3) ordinary tag names (denoted as ●). A tree structure including these notations is illustrated in Figure 3 [19], where 'nm' stands for 'name' and 'tel' stands for 'telephone number'.

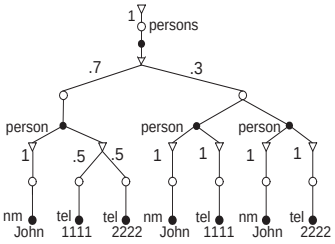

Fig. 3. A probabilistic tree simulating part of an XML document

Since the authors in the paper did not provide the actual XML structure for the example (or any other examples) to explicitly show how these types of tags are represented, we created an XML document for this example based on our own understanding as demonstrated in Figure 4 left. As we can see, there is lot of redundant information in this XML document, such as all the tags related to `possible values` are not strictly required, since a `possible` tag will always sit between a `probability` tag and a normal tag.

This XML structure can be easily rewritten into the XML format designed in [31] as shown in Figure 4 right which is more straightforward. However, an XML document written in the latter format cannot be converted into the structure of the former, since the former only deals with *strict* probability distributions, which is referred to as *mutually-exclusive* in the latter and does not discuss how the *non-mutually-exclusive* probabilities should be processed.

4 Mass function based XML

If we examine the content of Figure 3 in more detail, we will find that it actually reveal the information about different combinations of a name and a telephone number. The left branch (with a probability value 0.7) says that the probability of a person with name `John` having a telephone number either 1111 or 2222 is 0.7. The right hand branch says that there could be two persons with the same name `John` having telephone numbers 1111 and 2222 respectively and the probability of this happening is 0.3. Sometimes for a situation where different scenarios of combinations of values have to be considered, probability distributions are not adequate since a belief maybe associated with a subset of scenarios, rather than with each single scenario. To address this problem, Dempster-Shafer theory of evidence is deployed in [30] where mass values are assigned to subsets of scenarios of combinations of values. Let us look at an example taken from the paper as illustrated in Figure 5.

```
⟨probability⟩
 ⟨prob value = "1.0"⟩
  ⟨possible values⟩
   ⟨persons⟩
    ⟨probability⟩
     ⟨prob value = "0.7"⟩
      ⟨possible values⟩
       ⟨person⟩
        ⟨probability⟩
         ⟨prob value = "1.0"⟩
          ⟨possible values⟩
           ⟨personName⟩John⟨personName⟩
          ⟨/possible⟩
         ⟨/prob⟩

        ⟨/probability⟩
        ⟨probability⟩
         ⟨prob value = "0.5"⟩
          ⟨possible values⟩
           ⟨telNumber⟩1111⟨telNumber⟩
          ⟨/possible⟩
         ⟨/prob⟩
         ⟨prob value = "0.5"⟩
          ⟨possible values⟩
           ⟨telNumber⟩2222⟨telNumber⟩
          ⟨/possible⟩
         ⟨/prob⟩
        ⟨/probability⟩
       ⟨/person⟩
      ⟨/possible⟩
     ⟨/prob⟩
    ⟨/probability⟩
    ⟨probability⟩
      ⋮
    ⟨/probability⟩
   ⟨/persons⟩
  ⟨/possible⟩
 ⟨/prob⟩
⟨/probability⟩
```

```
⟨persons⟩
 ⟨Dist type = "m − e"⟩
  ⟨Val prob = "0.7"⟩
   ⟨person⟩
    ⟨name⟩John⟨/name⟩
    ⟨tel⟩
     ⟨Dist type = "m − e"⟩
      ⟨Val prob = "0.5"⟩1111⟨/Val⟩
      ⟨Val prob = "0.5"⟩2222⟨/Val⟩
     ⟨/Dist⟩
    ⟨/tel⟩
   ⟨/person⟩
  ⟨/Val⟩
  ⟨Val prob = "0.3"⟩
     ⋮
  ⟨/Val⟩
 ⟨/Dist⟩
⟨/persons⟩
```

Fig. 4. Two XML documents containing the probabilistic information given in Figure 3 where "m-e" stands for "mutually-exclusive"

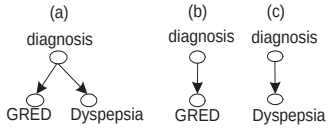

Fig. 5. Scenarios of history of a patient's diagnoses. Scenario (a) has two results, (b) and (c) each has one result.

The example shows how to model the history of diagnoses of a patient. Since this information may be collected over time through patient records and conversations with the patient, precise information as exactly what diagnoses the patient has had can be ambiguous in some cases. Give a particular patient with two possible diagnoses, GRED and Dyspepsia, the three scenarios in Figure 5 show the three possible situations (that can be modelled as three segments of an XML document) as what the patient might actually have had. In a case of certainty, e.g., the diagnosis is GRED, then the other two scenarios are redundant, so only one piece of information is useful and only one segment of XML document is needed. In a case of uncertainty, this collection of scenarios (called data forests in [30]) can be assigned beliefs in terms of mass functions[1] to tell which scenario(s) is more likely. For instance, the following mass functions are all possible where elements a, b, c in these mass functions refer to the scenarios in Figure 5. Element I in mass function m_4 says that if the two diagnoses, GERD and Dyspepsia, are both wrong, then the diagnosis can be something else. This element is added in [30] to guarantee that the set of values is always exhaustive.

$$
\begin{array}{l}
m_1(\{a\}) = 0.2,\ m_1(\{b\}) = 0.4, \qquad m_1(\{c\}) = 0.4; \\
m_2(\{a\}) = 0.4,\ m_2(\{a, b, c\}) = 0.6; \\
m_3(\{a\}) = 0.4,\ m_3(\{b, c\}) = 0.6; \\
m_4(\{a\}) = 0.4,\ m_4(I) = 0.6.
\end{array}
$$

Mass function m_1 is actually a probability distribution, so this distribution and the three scenarios can be modelled using the XML formats in both [19] and [31]. The other three mass functions can also be modelled using the format in [19], since this structure allows replicates of segments. However, it is not clear if this structure can support the manipulation of mass functions, if there are multiple mass functions available in a single XML document. As for the format provided by [31], it is unclear if it can be used to represent these mass distributions, especially the ones like m_2, since $\{a\}$ and $\{a, b, c\}$ are certainly not mutually exclusive, but the sum of their mass values has to be one. Furthermore, the calculations approach proposed in the paper will not be adequate for mass distributions either.

5 Hybrid uncertainties in XML

As we have seen above that both probabilistic and belief information can be present in XML documents. It is also possible that such pieces of information co-exist in a single XML file since information can be accumulated over time. A single uncertainty reasoning oriented XML is not sufficient to accommodate both types of information.

A logic based fusion rule technique, which can model various types of uncertain information within different segments of an XML file has been developed in [23–25] to address the needs of heterogeneity in uncertain information. Uncertainty can be modelled in either probability theory, belief function theory [40], or possibility theory [16].

[1] A mass function is defined as: $m : 2^S \rightarrow [0, 1]$ such that $m(A) \geq 0$ and $\Sigma_{A \subseteq S} m(A) = 1$, where S is a set of mutually exclusive and exhaustive values to a variable.

We consider two types of uncertainty described in [23] here, probability values and mass functions in DS theory. The formal modelling approach to representing these two types of uncertainty is given in the following two definitions.

A segment of XML document is a **probability-valid component** if the segment satisfies the following structure constraint.

$\langle$`probability`$\rangle\sigma_1, .., \sigma_n\langle/$`probability`$\rangle$ where $\sigma_i \in \{\sigma_1, .., \sigma_n\}$ is of the form $\langle$`prob value` $= \kappa\rangle\ \phi\langle/$`prob`$\rangle$ where $\kappa \in [0, 1]$ and ϕ is a textentry.

A segment of XML document is a **belfunction-valid component** if the segment satisfies the following structure constraint.

$\langle$`belfunction`$\rangle\sigma_1, .., \sigma_n\langle/$`belfunction`$\rangle$ where $\sigma_i \in \{\sigma_1, .., \sigma_n\}$ is of the form $\langle$`mass value` $= \kappa\rangle\sigma_1^i, ..., \sigma_m^i\langle/$`mass`$\rangle$ and for each $\sigma_j^i \in \{\sigma_1^i, .., \sigma_m^i\}$, σ_j^i is of the form $\langle$`massitem`$\rangle\phi\langle/$`massitem`$\rangle$ where $\kappa \in [0, 1]$ and ϕ is a textentry.

All textentries in the above two definitions are elements of a pre-defined set S in the background knowledgebase. We also require that $\Sigma_i\kappa_i = 1$ for both cases to preserve the constrains in both theories.

Let us take prostate cancer prediction and diagnosis as an example to see how to use the above structures to mode uncertain information. Higher Prostate Specific Antigen (PSA) value through a blood test can flag the possibility of cancer. However, this method is subject to inaccuracy, due to the fact that a higher PSA value can be influenced by many other factors, such as prostate inflammation and horse riding, before taking the blood sample. In general, this method is about 70% accurate in cancer diagnosis (http://medic.med.uth.tmc.edu/ptnt/00000390.htm). This high level summary can be represented in an XML document as shown in Figure 6.

Now assume that Patient A has a PSA value 12. This value can be fed into the above structure to get initial diagnosis which says that this patient has cancer with probability 0.65, Since this diagnosis is not 100% accurate, the reliability factor has to be considered too. In [23], a probability-valid segment can be converted into a belief function-valid segment and in [25] a belief function-valid segment can be processed to integrate reliability factor into mass functions. Therefore, using the information in Figure 6 coupled with PSA=12 for Patient A, the final diagnosis gives $m(\mathsf{Cancer}) = 0.455, m(\mathsf{NoCancer}) = 0.245, m(\mathsf{Cancer}, \mathsf{NoCancer}) = 0.3$ and the XML segment is

```
⟨belfunction⟩
  ⟨mass value = "0.455"⟩
    ⟨massitem⟩Cancer⟨/massitem⟩
  ⟨/mass⟩
  ⟨mass value = "0.245"⟩
    ⟨massitem⟩NoCancer⟨/massitem⟩
  ⟨/mass⟩
  ⟨mass value = "0.3"⟩
    ⟨massitem⟩Cancer⟨/massitem⟩
    ⟨massitem⟩NoCancer⟨/massitem⟩
  ⟨/mass⟩
⟨/belfunction⟩
```

In the fusion rule approach, non-leaf level uncertain information, such as probability values, is represented as reliability values, in contrast to both cases in [19] and [31], where the latter two use probability notations throughout. As demonstrated in [25], calculations of the final degree of probability of a value belonging to a tag produce the same result from [25] and [31] given the same XML information, eventhough,

```
⟨report⟩
  ⟨prostate cancer prediction⟩
   ⟨reliability = "0.7"⟩
    ⟨author⟩unknown ⟨/author⟩
    ⟨title⟩Prostatic Specific Antigen Screening Test⟨/title⟩
    ⟨url⟩http : //medic.med.uth.tmc.edu/ptnt/00000390.htm⟨/url⟩
     ⟨PSA range = "0.0 − 3.9"⟩
      ⟨conclusion⟩NoCancer⟨/conclusion⟩
     ⟨/PSA⟩
     ⟨PSA range = "4.0 − 9.9"⟩
      ⟨conclusion⟩
       ⟨probability⟩
        ⟨prob value = "0.22"⟩Cancer⟨/prob⟩
        ⟨prob value = "0.78"⟩NoCancer⟨/prob⟩
       ⟨/probability⟩
      ⟨/conclusion⟩
     ⟨/PSA⟩
     ⟨PSA range > "10.0"⟩
      ⟨conclusion⟩
       ⟨probability⟩
        ⟨prob value = "0.65"⟩Cancer⟨/prob⟩
        ⟨prob value = "0.35"⟩NoCancer⟨/prob⟩
       ⟨/probability⟩
      ⟨/conclusion⟩
     ⟨/PSA⟩
   ⟨/reliability⟩
  ⟨/prostate cancer prediction⟩
⟨/report⟩
```

Fig. 6. An XML document with uncertain information

the uncertain information is modelled differently. In this sense, these two approaches are equivalent in terms of modelling information. However the approach in [23, 25] is better than that in [31] in respect to merging information, which is not considered in [31]. The main focus of [19] is to merge multiple XML documents with no or limited uncertain information to start with and to produce the merged document with probabilistic values. The manipulation of final probabilities seems to be very simple (which was not explicitly discussed in the paper). Since there is little background knowledge to guide the merge process, almost all combinations of branches (called *possible worlds* in [19], which are usually near to the leaf level) from two XML documents need to be considered, and various forms of structures of XML document can be produced as a result of a simple merge. This overwhelmingly complex procedure can be simplified with the adequate use of background knowledge to assess which two segments may contain related information to make the merging more sensible, as used in [23]. Therefore, [23] is better than [19], especially for merging complex XML documents with highly relevant information. In terms of modelling information, the former is simpler than the latter too, as shown in Figure 7 which models the same information as that in Figure 4 left. The apparent structural differences among [19, 23, 25, 31] is being investigated in which XSLT is used as part of a tool to transform one XML document into the format of another XML document.

The definition of belfunction-valid component on leaf-node can be easily extended to non-leaf nodes, e.g., to branches as suggested in [30]. By extending leaf-level bel-

```
⟨persons⟩
 ⟨reliability value = "0.7"⟩
  ⟨name⟩John⟨/name⟩
  ⟨tel⟩
   ⟨probability⟩
    ⟨prob value = "0.5"⟩1111⟨/prob⟩
    ⟨prob value = "0.5"⟩2222⟨/prob⟩
   ⟨/probability⟩
  ⟨/tel⟩
 ⟨/reliability⟩
 ⟨reliability⟩
    ⋮
 ⟨/reliability⟩
⟨/persons⟩
```

Fig. 7. An XML document for the same information as in Figure 4

function component to non-leaf level nodes, coupled with assistant fusion rules, the rest of the fusion rule technique can be directly applied.

6 Conclusion

In this short survey paper, we have examined the current state of the art of modelling and merging uncertain information presented in XML documents. As XML is being increasingly used on the Web as a standard for data storage and exchange, modelling uncertain and incomplete information as well as merging these pieces of information have become an important and urgent issue.

Various approaches proposed so far have different advantages and limitations, some approaches focused on modelling, the others addressed merging uncertain information. The only approach that considered both is [23] which also has the ability to deploy hybrid uncertain reasoning mechanisms. It seems that the fusion rule technique can provide a formal platform for addressing these issues and has the potential to standardize the various proposals of modelling uncertain information in XML available so far.

A collection of papers that we did not discuss in this paper is on using probability intervals to model uncertainty in XML (e.g.,[15, 20, 21, 46]). In [15], the idea of dealing with probabilities (and intervals) and semistructured data was first proposed. This model was extended to XML structure in [46]. Although the apparent format is XML-based, the query manipulation is very much traditional database oriented using algebras. Similarly, a semistructured instance in [21] can be taken as a branch of an XML document. The manipulations of probabilities, either intervals ([21]) or point based ([20]) are also database oriented.

References

1. SCI: http://www.scientio.com
2. TEI: http://www.tei-c.org/P4X/CE.html

3. PMML: http://www.dmg.org/index.html
4. SAGE: http://cgap.nci.nih.gov/SAGE
5. A Al-Ani and M Deriche. A new technique for combining multiple classifiers using the Dempster-Shafer theory of evidence. Journal of Artificial Intelligence Research 17:333-361, 2002.
6. D Burdick, P Deshpande, and T Jayram. OLAP over uncertain and imprecise data. *Proc. of VLDB'05*:970-981, 2005.
7. D Barbara, H Garcia-Molina, and D Porter. The management of probabilistic data. *IEEE Trans. on Knowledge and Data Engineering*, 4(5):487-502, 1992.
8. E Begley and C Howard-Reed. The application of MatML to contaminant emissions data. *ASTM STANDARDIZATION NEWS*. October 2005.
9. P Ceravolo, E Damiani and B Oliboni. Fuzzy technique for metadata construction. *Proc. of IPMU'04*:1019-1026. 2004.
10. J Cowie and W Lehnert. Information extraction. *Comm. of the ACM*, 39:81–91, 1996.
11. C Combi, B Oliboni, and R Rossato. Merging multimedia presentations and semi-structured temporal data: a graph-based model and its application to clinical information. *Artificial Intelligence in Medicine*, 2005.
12. R Cavallo, and M Pittarelli. The theory of probabilistic databases. *Proc. of VLBD'87*:71-81, 1987.
13. R Chirico, M Frenkel, V Diky, K Marsh, and R Wilhoit. ThermoML - an XML-based Approach for Storage and Exchange of Experimental and Critically Evaluated Thermophysical and Thermochemical Property Data, 2, Uncertainties. *Journal of Chemical and Engineering Data*, 48:1344-1359, 2003.
14. D Draper, A HaLevy, and D Weld. The Nimble XML Data Integration System. *Proc. of the 17th International Conference on Data Engineering*:155-160, 2001.
15. A Dekhtyar, J Goldsmith, and S Hawkes. Semistructured probabilistic databases. *Proc. of Statistical and Scientific Databases Management Systems*:36-45, 2001.
16. D Dubois and H Prade. *Possibility theory: An approach to the computerized processing of uncertainty*. Plenum Press, 1988.
17. N Dalvi and D Suciu. Answering queries from statistical and probabilistic views. *Proc. of VLDB05*:805-816, 2005.
18. N Elhadad, M Kan, J Klavans, and K McKeown. Customization in a unified framework for summarizing medical literature. *Artificial Intelligence in Medicine* 33,179-198, 2005.
19. M van Keulen, A de Keijzer and W Alink. A probabilistic XML approach to data integration. *Proceedings of ICDE'05*, 459-470, 2005.
20. E Hung, L Getoor, and V Subrahmanian. PXML: A Probabilistic Semistructured Data Model and Algebra. *Proc. of International Conference on Data Engineering (ICDE03)*:467-480, 2003.
21. E Hung, L Getoor, and V Subrahmanian. Probabilistic Interval XML. *International Conference on Database Theory (ICDT'03)*:361-377, 2003.
22. M Holi and E Hyvonen. A method for modelling uncertainty in Semantic web taxonomies. *Proc. of WWW'04*:296-297, 2004.
23. A Hunter and W Liu. Fusion rules for merging uncertain information. *Information Fusion* (in press), 2005.
24. A Hunter and W Liu. Merging uncertain information with semantic heterogeneity in XML. *Knowledge and Information Systems* (in press), 2005.
25. A Hunter and W Liu. A logical reasoning framework for modelling and merging uncertain semi-structured information. *Modern Information Processing: From Theory to Applications*, B. Bouchon-Meunier, G. Coletti and R.R. Yager (eds.), Elsevier (in press), 2005.
26. A Hunter. Logical fusion rules for merging structured news reports. *Data and Knowledge Engineering*, 42:23–56, 2002.

27. A Hunter and R Summerton. Fusion rules for context-dependent aggregation of structured news reports. *Journal of Applied Non-classical Logic* 14(3):329-366, 2004.
28. P Lehti and P Fankhauser. XML data integration with OWL: experiences and challenges. *Proc. of Symposium on Applications and the Internet (SAINT04)*:160-167, 2004.
29. T Lukasiewicz. Probabilistic description logic programs *Proc. of the 8th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU05)*:737-749, 2005
30. M Magnani and D Montesi. A model for imperfect XML data based on Dempster-Shafer's theory of evidence. Technical report, Department of Computer Science, University of Bologna, 2005.
31. A Nierman and H Jagadish. ProTDB: Probabilistic data in XML. In *Proc. of VLDB'02*, LNCS2590: 646–657. Springer, 2002.
32. H Nottelmann and N Fuhr. pDAML+OIL: A probabilistic extension to DAML+OIL based on probability datalog. *Proc. of IPMU'04*:227-234, 2004.
33. M Orchard. On modelling location uncertainty in images. *Proc. of International Conference on Image Processing*, Vol 1, 2001.
34. T Pankowski and E Hunt. Data merging in life science data integration systems. *Intelligent Information Systems, Advances in Soft Computing*, Springer, 2005.
35. S Philippe and J Köhler. Using XML technology for ontology-based semantic integration of life science databases. *IEEE Trans. on Information Technology in Bioinformatics* 8(2):154-160, 2004.
36. X Pennec and J Thirion. A Framework for Uncertainty and Validation of 3D Registration Methods based on Points and Frames. *International Journal of Computer Vision*, 25(3):203–229, 1997.
37. D Radev, W Fan, H Qi, H Wu and A Grewal. Probabilistic question answering on the Web. *proc. of WWW'02*:408-419, 2002.
38. E Santos. Unifying time and uncertainty for diagnosis. *Journal of Experimental and Theoretical Artificial Intelligence*, 8: 75-94, 1996.
39. L Seligman and A Rosenthal. XML's impact on databases and data sharing. *IEEE Computer* 34:59-61, June, 2001.
40. G Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.
41. U Straccia. Reasoning within fuzzy description logics *J. of Artificial Intelligence Research*:137-166, 2001.
42. G Stoilos, G Stamou, V Tzouvaras, J Pan, and I Horrocks. A fuzzy extension of SWRL. *Proc. of W3C Workshop on Rule Languages for Interoperability*, 2004.
43. G Stoilos, G Stamou, V Tzouvaras, J Pan, and I Horrocks. A fuzzy description logic for multimedia knowledge representation. *Proc. of the International Workshop on Multimedia and the Semantic Web*:12-19, 2005.
44. H Wang, M Zhang, D Xu, D Zhang. A Framework of Fuzzy Diagnosis. *IEEE Transactions on Knowledge and Data Engineering*, 16 (12): 1571-1582, 2004.
45. L Zamboulis. XML Data Integration By Graph Restructuring. *Proc. BNCOD'04*:57-71, LNCS 3112, July 2004.
46. W Zhao, A Dekhtyar, and J Goldsmith. Representing probabilistic information in XML. *Technical Report 770-03*, Department of Computer Science, University of Kentucky, 2003.
47. B Zupan, J Demsar, M Katten, M Ohori, M Graefen, M Bojanec, and R Beck. Orange and decisions-at-hand: bridging predictive data mining and decision support. *Proc. of ECML/PKDD'01 workshop on Integrating Aspects of Data Mining Decision Support and Meta-Learning*:151–162, September 2001.