Logical reasoning with multiple granularities of uncertainty in semi-structured information

Anthony Hunter
Department of Computer Science
University College London
Gower Street, London WC1E 6BT, UK
a.hunter@ucl.ac.uk

Weiru Liu
School of Computer Science
Queen's University of Belfast
Belfast, Co. Antrim, BT7 1NN, UK
w.liu@qub.ac.uk

Abstract

Semi-structured information in XML can be merged in a logic-based framework [3, 4]. In this paper we extend this approach to modelling and merging uncertain information that is defined either on textentries or at different levels of granularities with XML textentries, as well as to modelling and reasoning with XML documents that contains semantic heterogeneous uncertain information on more complex elements in XML subtrees.

Keywords: DS theory, logic-based information fusion, uncertain semi-structured information.

1 Introduction

We use XML documents to represent semi-structured information such as structured scientific knowledge (SSK). Each SSK report describes information in one or more scientific datasources (such as one or more journals, databases of empirical results, etc). The format of an SSK report is an XML document where the tagnames provide the semantic structure and coherence to the document and the textentries (i.e. leaves) are restricted to (1) individual words or simple phrases from a scientific nomenclature/terminology and (2) individual numerical values with units.

An SSK report is intended to help scientists understand the contents of a datasource. Each SSK report contains **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). The summaritive information describes the information provided by the authors of the datasource, and the evaluative information describes the information provided by the users or authors of the datasource. Each SSK report can be constructed by hand, by information extraction systems, (e.g. [1]), or as by-product of querying and analysing scientific databases.

Whilst SSK reports are a useful resource for representing information in applications such as bioinformatics and e-science, there is a pressing need to develop tools to analyse and integrate them. To address this need we have been developing a logic-based formalism that supports context-dependent representing and reasoning with the uncertainty in this information.

In our approach, each SSK report is regarded as a tree and this can isomorphically be represented as a logical term: Each tagname is a function symbol, and each textentry is a constant symbol. Furthermore, subtrees of an SSK can be isomorphically represented as subterms in logic. In this way, the information in each SSK report can be captured in a logical language. We have also defined a range of predicates, in a Prolog knowledgebase, that capture useful relationships between SSK reports, and so a set of them can then be analysed or merged as Prolog queries to a Prolog knowledgebase. In this way, a query to merge some SSK reports can be handled by recursive calls to Prolog to merge the subtrees in the SSK reports. This gives a context-dependent logic-based approach to merging that is sensitive to the uncertain information in the SSK reports and to the background knowledge in the Prolog knowledgebase.

In this paper, we propose approaches to modelling and merging uncertain information, in the form of mass functions in the Dempster-Shafer theory of evidene (DS theory) ([8]), on textentries, as demonstrated in Example 1. We also investigate situations as illustrated in Example 2, where the first XML document shows that pieces of evidence can be given at different levels of granularities of the same concept represented by a textentry, and the second one reveals that a mass function can be assigned to a subtree of an XML document. We will provide formal definitions on how to model and reason with these types of XML structure, as well as how to merge multiple SSK reports with such uncertain information.

We will proceed as follows. In Section 2, we present formal definitions of logical representations of XML documents, review the basics of DS theory, and provide formal definitions of modelling and merging uncertain information in SSK reports in the form of mass functions on the same textentry of two XML documents. In Section 3, we consider propagating and merging uncertain information at different levels of granularities. In Section 4 we investigate methods of reasoning with uncertain information on subtrees. Finally, we compare our work with related research in the final section and provide conclusions.

Example 1 *Consider the following two SSK reports which are for the same area being explorated. Both of them define a mass function on the textentry* `deposit`. *A fusion predicate defined later in Section 2 generates a merged SSK report with the combined mass function.*

```
⟨report⟩
  ⟨source⟩ Experiment1 ⟨/source⟩
  ⟨date⟩ 19/3/02 ⟨/date⟩
  ⟨location⟩ NorthSea ⟨/location⟩
  ⟨deposit⟩
    ⟨belfunction⟩
      ⟨mass value = “0.4”⟩
        ⟨massitem⟩water⟨/massitem⟩
        ⟨massitem⟩oil⟨/massitem⟩
      ⟨/mass⟩
      ⟨mass value = “0.6”⟩
        ⟨massitem⟩gas⟨/massitem⟩
      ⟨/mass⟩
    ⟨/belfunction⟩
  ⟨/deposit⟩
⟨/report⟩

⟨report⟩
  ⟨source⟩ Experiment2 ⟨/source⟩
  ⟨date⟩ 19 March 2002 ⟨/date⟩
  ⟨location⟩ NorthSea ⟨/location⟩
  ⟨deposit⟩
    ⟨belfunction⟩
      ⟨mass value = “0.2”⟩
        ⟨massitem⟩water⟨/massitem⟩
      ⟨/mass⟩
      ⟨mass value = “0.8”⟩
        ⟨massitem⟩gas⟨/massitem⟩
      ⟨/mass⟩
    ⟨/belfunction⟩
  ⟨/deposit⟩
⟨/report⟩
```

The merged result provides a new XML document with combined mass function as shown below.

```
⟨report⟩
  ⟨source⟩ Exp1 and Exp2 ⟨/source⟩
  ⟨date⟩ 19/3/02 ⟨/date⟩
  ⟨location⟩ NorthSea ⟨/location⟩
  ⟨deposit⟩
    ⟨belfunction⟩
      ⟨mass value = “0.143”⟩
        ⟨massitem⟩water⟨/massitem⟩
      ⟨/mass⟩
      ⟨mass value = “0.857”⟩
        ⟨massitem⟩gas⟨/massitem⟩
      ⟨/mass⟩
    ⟨/belfunction⟩
  ⟨/deposit⟩
⟨/report⟩
```

Here in this and subsequent examples, we use some simplified data from the petroleum exploration domain. The main purpose of petrolem exploration is to analysis qualitatively and calculate quantitatively the well logging data in order to predict the possible deposits in some areas. The well logging data are digital records which can reflect the underground physical features, for instance, electronic resistance, micro-electrode resistance, and natural gamma ray, etc. They are collected by well *logging equipment inside the well from the ground level to some depth underground. The whole depth from the ground level to the bottom of the well is divided into layers based on the digital data collected and the values of these physical features can give indications of layers with possible deposits. The first two XML documents in Example 1 show how an expert can predict a possible deposit of a particular layer, by examining the digital data of the layer. Since equipment used is subject to noise and inaccuracy, multiple experiments are needed in order to make an accurate prediction. Furthermore, the general analysis of the broader area of the physical features of the location often provides some additional information for predication. The 2nd report in Example 2 shows how this knowledge can be simplified and coded.*

Example 2 *The first XML report below gives a pieces of evidence in the form of mass function on the frame* `deposit` *with values* {`liquid`, `solid`}, *in contrast with the two mass functions given in Example 1, where the frame* `deposit` *has values* {`water`, `oil`, `gas`, `sand`, `stone`}. *The second XML report includes a mass function defined on a subtree, where the mass values are assigned to elements not from a single frame but from multiple frames. To merge these two reports, one mass function has to be propagated onto the frame of another.*

```
⟨report⟩
  ⟨source⟩ Experiment3 ⟨/source⟩
  ⟨date⟩ 19/3/02 ⟨/date⟩
  ⟨location⟩ NorthSea ⟨/location⟩
  ⟨deposit⟩
    ⟨belfunction⟩
      ⟨mass value = “0.2”⟩
        ⟨massitem⟩liquid⟨/massitem⟩
      ⟨/mass⟩
      ⟨mass value = “0.8”⟩
        ⟨massitem⟩solid⟨/massitem⟩
      ⟨/mass⟩
    ⟨/belfunction⟩
  ⟨/deposit⟩
⟨/report⟩
```

```
⟨report⟩
  ⟨source⟩ General Knowledge ⟨/source⟩
  ⟨date⟩ 19/3/02 ⟨/date⟩
  ⟨location⟩ NorthSea ⟨/location⟩
    ⟨belfunction⟩
      ⟨mass value = “0.4”⟩
        ⟨deposit⟩water⟨/deposit⟩
        ⟨lithology⟩L1⟨/litology⟩
      ⟨/mass⟩
      ⟨mass value = “0.6”⟩
        ⟨deposit⟩gas⟨/deposit⟩
        ⟨lithology⟩L2⟨/lithology⟩
      ⟨/mass⟩
    ⟨/belfunction⟩
⟨/report⟩
```

2 Structured Scientific Knowledge

We now provide basic definitions for SSK, and the logical representation of them. We then introduce the basics of DS theory before considering how to represent and merge uncertain information in SSK.

Basic definitions: We use XML to represent SSK reports. So each SSK report is an XML document, but not vice versa, as defined below. This restriction means that we can easily represent each SSK report by a ground term in classical logic.

Def. 1 Structured Scientific Knowledge Report (SSK report): *If φ is a tagname (i.e an element name), and ϕ is textentry, then $\langle\varphi\rangle\phi\langle/\varphi\rangle$ is an SSK report. If φ is a tagname, ϕ is textentry, θ is an attribute name, and κ is an attribute value, then $\langle\varphi\ \theta = \kappa\rangle\phi\langle/\varphi\rangle$ is an SSK report. If φ is an tagname and $\sigma_1, ..., \sigma_n$ are SSK reports, then $\langle\varphi\rangle\sigma_1...\sigma_n\langle/\varphi\rangle$ is an SSK report.*

Clearly each SSK report is isomorphic to a tree with the non-leaf nodes being the tagnames and the leaf nodes being the textentries. This isomorphism allows us to give a definition for an *abstract term* of an SSK report.

Def. 2 Abstract term: *Each SSK report is isomorphic with a ground term (of classical logic) called an abstract term. This isomorphism is defined inductively as follows: (1) If $\langle\varphi\rangle\phi\langle/\varphi\rangle$ is an SSK report, where ϕ is a textentry, then $\varphi(\phi)$ is an abstract term that is isomorphic with $\langle\varphi\rangle\phi\langle/\varphi\rangle$; (2) If $\langle\varphi\ \theta = \kappa\rangle\phi\langle/\varphi\rangle$ is an SSK report, where ϕ is a textentry, then $\varphi(\phi, \kappa)$ is an abstract term that is isomorphic with $\langle\varphi\ \theta = \kappa\rangle\phi\langle/\varphi\rangle$; and (3) If $\langle\varphi\rangle\phi_1..\phi_n\langle/\varphi\rangle$ is an SSK report, and ϕ'_1 is an abstract term that is isomorphic with ϕ_1, ..., and ϕ'_n is an abstract term that is isomorphic with ϕ_n, then $\varphi(\phi'_1, .., \phi'_n)$ is an abstract term that is isomorphic with $\langle\varphi\rangle\phi_1..\phi_n\langle/\varphi\rangle$.*

Via this isomorphic relationship, we can refer to a branch of an abstract term by using the branch of the isomorphic SSK, and we can refer to a subtree of an abstract term by using the subtree of the isomorphic SSK.

Def. 1 describes how an XML document can be defined recursively starting from the simpliest one which has only one tag name and one value associated with the tag name. Def. 2 defines how a tree structure like XML document can be equally described as a logical term which also reflects the relationships between tag names and their values. For instance, XML information ⟨date⟩03/03/99⟨/date⟩ is denoted as date(03/03/99) in logics where 03/03/99 can be understood as the value of attribute date.

Basics of DS theory: DS theory has a commonly accepted advantage than probability theory in terms of assinging a proportion of an agent's belief to a subset of a set of possible values than only on singletons, and assigning any unspecified proportion of belief to the whole set. This is especially useful when the evidence supporting an agent's belief is not accurate or incomplete. Furthermore, multiple pieces of evidence can be accumulated over time on the same subject and these pieces of evidence should be combined/merged in some way in order to draw a conclusion out of them. Dempster's combination rule in DS theory provides a simple mechanism to achieve this objective. Due to these two advantages provided by DS theory, we have chosen DS theory to model, reason and merge uncertain information in SSK in the form of XML documents in this paper.

Let Ω be a finite set containing mutually exclusive and exhaustive solutions to a question. Ω is called a **frame of discernment**. A **mass function**, also called a **basic probability assignment**, captures the impact of a piece of evidence on subsets of Ω. A mass functions $m : \wp(\Omega) \rightarrow [0, 1]$ satisfies:

$$m(\emptyset) = 0 \text{ and } \Sigma_{A \subseteq \Omega}\, m(A) = 1$$

When $m(A) > 0$, A is referred to as a **focal element**. To obtain the total belief in a subset A, i.e. the extent to which all available evidence supports A, we need to sum all the mass assigned to all subsets of A. A **belief function**, $Bel : \wp(\Omega) \rightarrow [0, 1]$, is defined as:

$$Bel(A) = \Sigma_{B \subseteq A} m(B)$$

A **plausibility function**, $Pl : \wp(\Omega) \rightarrow [0, 1]$, is defined as follows

$$Pl(A) = 1 - Bel(\bar{A}) = \Sigma_{B \cap A \neq \emptyset}\, m(B)$$

Dempster's rule of combination below shows how two mass functions m_1 and m_2, on the same frame from independent sources, can be combined to produce a merged one.

$$m_1 \oplus m_2(C) = \frac{\Sigma_{A \cap B = C}\ (m_1(A) \times m_2(B))}{1 - \Sigma_{A \cap B = \emptyset}\ (m_1(A) \times m_2(B))}$$

Modelling uncertain information: In order to support the representation of uncertain information in SSK reports, we need some further formalization. First, we assume a set of tagnames that are reserved for representing uncertain information. Second, we assume some constraints on the use of these tags so that we can ensure they are used in a meaningful way with respect to DS theory.

Def. 3 *The tagnames* belfunction multiitem, mass, *and* massitem *are called* **reserved tagnames**.

Def. 4 *The SSK* ⟨belfunction⟩ $\sigma_1, .., \sigma_n$⟨/belfunction⟩ *is* **belfunction-valid** *iff for each* $\sigma_i \in \{\sigma_1, .., \sigma_n\}$ σ_i *is of the form* ⟨mass value $= \kappa$⟩$\sigma^i_1, ..., \sigma^i_m$⟨/mass⟩ *and for each* $\sigma^i_j \in \{\sigma^i_1, .., \sigma^i_m\}$, σ^i_j *is of the form* ⟨massitem⟩ϕ⟨/massitem⟩ *where* $\kappa \in [0, 1]$ *and* ϕ *is a textentry.*

The textentries in a belfunction-valid component are from a pre-defined frame. When two such components are available on the same tagname, the following procedure merges them using the Dempster's combination rule.

Def. 5 *Let the following be two belfunction-valid uncertainty components*

⟨belfunction⟩$\sigma^1_1, .., \sigma^1_p$⟨/belfunction⟩
⟨belfunction⟩$\sigma^2_1, .., \sigma^2_q$⟨/belfunction⟩

where

1. $\sigma^1_i \in \{\sigma^1_1, .., \sigma^1_p\}$ *is of the form* ⟨mass value $= \kappa^1_i$⟩ψ^1_i⟨/mass⟩
2. ψ^1_i *is of the form*

   ⟨massitem⟩$\phi^1_{i_1}$⟨/massitem⟩
   $\vdots$
   ⟨massitem⟩$\phi^1_{i_x}$⟨/massitem⟩
3. $\sigma^2_j \in \{\sigma^2_1, .., \sigma^2_q\}$ *is of the form* ⟨mass value $= \kappa^2_j$⟩ψ^2_i⟨/mass⟩
4. ψ^2_j *is of the form*

   ⟨massitem⟩$\phi^2_{j_1}$⟨/massitem⟩
   $\vdots$
   ⟨massitem⟩$\phi^2_{j_y}$⟨/massitem⟩

Let the **combined belfunction component** *be*

$$\langle\texttt{belfunction}\rangle\sigma_1,..,\sigma_s\langle/\texttt{belfunction}\rangle$$

where each $\sigma_k \in \{\sigma_1,..,\sigma_s\}$ *is of the form* $\langle\texttt{mass value} = \kappa_k\rangle\psi\langle/\texttt{mass}\rangle$ *and* $\kappa_k = \frac{\Sigma\kappa_i^1\times\kappa_j^2}{1-\kappa_\perp}$ *and* $\kappa_\perp = \Sigma\kappa_n^1\times\kappa_m^2$ *and*

1. ψ *is of the form*

$$\langle\texttt{massitem}\rangle\phi_1\langle/\texttt{massitem}\rangle$$
$$\vdots$$
$$\langle\texttt{massitem}\rangle\phi_z\langle/\texttt{massitem}\rangle$$

2. $\{\phi_1,\cdots,\phi_z\} = \{\phi^1_{i_1},\cdots,\phi^1_{i_x}\}\cap\{\phi^2_{j_1},\cdots,\phi^2_{j_y}\}$
3. $\sigma^1_n \in \{\sigma^1_1,..,\sigma^1_p\}$ *is of the form* $\langle\texttt{mass value} = \kappa^1_n\rangle\psi^1_n\langle/\texttt{mass}\rangle$
4. ψ^1_n *is of the form*

$$\langle\texttt{massitem}\rangle\phi^1_{n_1}\langle/\texttt{massitem}\rangle$$
$$\vdots$$
$$\langle\texttt{massitem}\rangle\phi^1_{n_v}\langle/\texttt{massitem}\rangle$$

5. $\sigma^2_m \in \{\sigma^2_1,..,\sigma^2_q\}$ *is of the form* $\langle\texttt{mass value} = \kappa^2_m\rangle\psi^2_m\langle/\texttt{mass}\rangle$
6. ψ^2_m *is of the form*

$$\langle\texttt{massitem}\rangle\phi^2_{m_1}\langle/\texttt{massitem}\rangle$$
$$\vdots$$
$$\langle\texttt{massitem}\rangle\phi^2_{m_w}\langle/\texttt{massitem}\rangle$$

7. $\{\phi^1_{n_1},\cdots,\phi^1_{n_v}\}\cap\{\phi^2_{m_1},\cdots,\phi^2_{m_w}\} = \emptyset$

A predicate $\texttt{Dempster}(\tau_1,\tau_2,X)$ is defined in Prolog to carry out the actural merge, where τ_1 and τ_2 are two belfunction-valid abstract terms and X is a logical variable. If we let τ_1 and τ_2 be the abstract terms for the first two XML documents in Ex. 1, then X represents the merged abstract term isomorphic to the third XML document in the example.

The above definitions define how to model amd merge mass functions at textentries, we now provide the definition for subtrees.

Def. 6 *The SSK* $\langle\texttt{belfunction}\rangle$ $\sigma_1,\ldots,\sigma_n$ $\langle/\texttt{belfunction}\rangle$ *is* **subtree-belfunction-valid** *iff for each* $\sigma_i \in \{\sigma_1,..,\sigma_n\}$ σ_i *is of the form* $\langle\texttt{mass value} = \kappa\rangle$ $\sigma^i_1,...,\sigma^i_m$ $\langle/\texttt{mass}\rangle$ *and for each* $\sigma^i_j \in \{\sigma^i_1,..,\sigma^i_m\}$, σ^i_j *is of the form* $\langle\texttt{multiitem}\rangle$ $\langle\psi^i_{j1}\rangle\phi^i_{j1}\langle/\psi^i_{j1}\rangle$, $\ldots$, $\langle\psi^i_{jl}\rangle\phi^i_{jl}\langle/\psi^i_{jl}\rangle$ $\langle/\texttt{multiitem}\rangle$, *and* $\kappa\in[0,1]$, *where* ψ^i_{jt} *are tagnames, and* ϕ^i_{jt} *are textentries.*

Any two tagnames from within a ($\langle\texttt{multiitem}\rangle$, $\langle/\texttt{multiitem}\rangle$) pair must be distinct and their corresponding textentries are from different sets containing mutually exclusive and exhaustive values. Either a belfunction-valid SSK report or a subtree-belfunction- valid SSK report is called an **uncertain component**.

Example 3 *The following is a subtree-belfunction-valid uncertain component providing a mass function on pairs of values from two related sets* deposit *and* lithology.

```
⟨belfunction⟩
  ⟨mass value = “0.4”⟩
    ⟨multiitem⟩
      ⟨depost⟩water⟨/deposit⟩
      ⟨lithology⟩L1⟨/lithology⟩
    ⟨/multiitem⟩
    ⟨multiitem⟩
      ⟨deposit⟩oil⟨/deposit⟩
      ⟨lithology⟩L3⟨/lithology⟩
    ⟨/multiitem⟩
  ⟨/mass⟩
  ⟨mass value = “0.6”⟩
    ⟨multiitem⟩
      ⟨deposit⟩gas⟨/deposit⟩
      ⟨lithology⟩L2⟨/lithology⟩
    ⟨/multiitem⟩
  ⟨/mass⟩
⟨/belefunction⟩
```

3 Merging Uncertain Information on Textentries with Compatible Frames

When two mass functions are not given on the same frame, they cannot be combined directly, rather one mass function has to be progagated to the frame holding another mass function. This is done through **compatibility mappings**

Let Ω_1 and Ω_2 be two frames of discernment and Γ be a mapping function $\Gamma : \Omega_1 \rightarrow 2^{\Omega_2}$. When the following conditions hold, Ω_2 is called a **refinement** of Ω_1 and Ω_1 is called a **coarsening** of Ω_2.

(1) $\Gamma(\phi) = S_\phi \neq \emptyset$, for all $\phi \in \Omega_1$;
(2) $\Gamma(\phi_i) \neq \Gamma(\phi_j)$, when $i \neq j$;
(3) $\cup_{\phi\in\Omega_1}\Gamma(\phi) = \Omega_2$.

Example 2 gives a mass function on frame $\Omega_1 = \{\texttt{liquid},\texttt{solid}\}$ and Example 1 gives two mass functions on $\Omega_2 = \{\texttt{water},\texttt{oil},\texttt{gas},\texttt{sand},\texttt{stone}\}$ where Ω_2 is a refinement of Ω_1, if we define the mapping function Γ as

$$\Gamma(\texttt{liquid}) = \{\texttt{water}, \texttt{oil}, \texttt{gas}\},$$
$$\Gamma(\texttt{solid}) = \{\texttt{sand}, \texttt{solid}\}.$$

Let Ω_2 be a refinement of frame Ω_1 and m_{Ω_1} be a mass function on Ω_1. Function m_{Ω_2} defined below is a mass function on Ω_2.

$$m_{\Omega_2}(B) = m_{\Omega_1}(A) \texttt{ where } B = \bigcup \Gamma(\phi), \forall \phi \in A \tag{1}$$

Equally, Let Ω_1 be a coarsening of frame Ω_2 with mapping function Γ', and m_{Ω_2} be a mass function on Ω_2. Function m_{Ω_1} defined below is a mass function on Ω_1.

$$m_{\Omega_1}(B) = \Sigma_A m_{\Omega_2}(A) \texttt{ where } B = \bigcup \Gamma'(\phi), \forall \phi \in A \tag{2}$$

Generally, we can describe two compatible relations as follows. Let Ω_1 and Ω_2 be two frames of discernment containing possible values to two distinct but related questions Q_1 and Q_2. Let Γ be a mapping function $\Gamma : \Omega_1 \rightarrow 2^{\Omega_2}$ where the mapping function Γ defines that whenever ϕ_i^1 is the true answer to question Q_1 then the true answer to the question Q_2 must be one of the elements in $\Gamma(\phi_i^1) \neq \emptyset$, and for every $\phi_j^2 \in \Omega_2$, there exists at least one ϕ_i^1 such that $\phi_j^2 \in \Gamma(\phi_i^1)$. Then frames Ω_1 and Ω_2 are said to be **compatible**. Mapping Γ is referred to as a **compatibility mapping** [5, 6]. Equally, a compatibility mapping can be defined from Ω_2 to Ω_1. A refinement (or coarsening) mapping is a special case of compatibility mapping.

For instance, different *deposit* possess different features such as their *lithologies*. The relationship between `deposit` and `lithology` can be established through a mapping Γ as

$$\Gamma(\texttt{water}) = \{\texttt{L1}, \texttt{L2}\}$$
$$\Gamma(\texttt{oil}) = \{\texttt{L3}, \texttt{L4}\}$$
$$\Gamma(\texttt{gas}) = \{\texttt{L2}, \texttt{L5}, \texttt{L6}\}$$
$$\Gamma(\texttt{sand}) = \{\texttt{L8}, \texttt{L9}\}$$
$$\Gamma(\texttt{stone}) = \{\texttt{L7}, \texttt{L8}\}$$

Let Ω_1 and Ω_2 be two related frames with a compatibility mapping Γ. Let m_{Ω_1} be a mass function on Ω_1. Then function m_{Ω_2} defined below is a mass function on Ω_2.

$$m_{\Omega_2}(B) = \Sigma_A m_{\Omega_1}(A) \texttt{ where } B = \bigcup \Gamma(\phi), \forall \phi \in A \tag{3}$$

All these three equations can be proved easily (e.g., [8]). We now provide the procedure to generate a mass function from another give two compatible frames.

Def. 7 *Let the following be a belfunction-valid uncertainty component*

$$\langle\texttt{belfunction}\rangle\sigma_1^1, .., \sigma_p^1\langle/\texttt{belfunction}\rangle$$

where

1. $\sigma_i^1 \in \{\sigma_1^1, .., \sigma_p^1\}$ *is of the form* $\langle\texttt{mass value} = \kappa_i^1\rangle\psi_i^1\langle/\texttt{mass}\rangle$
2. ψ_i^1 *is of the form*

$$\langle\texttt{massitem}\rangle\phi_{i_1}^1\langle/\texttt{massitem}\rangle$$
$$\vdots$$
$$\langle\texttt{massitem}\rangle\phi_{i_x}^1\langle/\texttt{massitem}\rangle$$

Let the frame associated with it be Ω_1 which has a compatibility mapping Γ with another frame Ω_2.

Let the propagated belfunction-valid component on Ω_2 be

$$\langle\texttt{belfunction}\rangle\sigma_1^2, .., \sigma_q^2\langle/\texttt{belfunction}\rangle$$

where each $\sigma_k \in \{\sigma_1^2, .., \sigma_q^2\}$ is of the form $\langle\texttt{mass value} = \kappa_k^2\rangle\psi_k^2\langle/\texttt{mass}\rangle$ and $\kappa_k^2 = \Sigma_i \kappa_i^1$ and

1. ψ_k^2 *is of the form*

$$\langle\texttt{massitem}\rangle\phi_1^2\langle/\texttt{massitem}\rangle$$
$$\vdots$$
$$\langle\texttt{massitem}\rangle\phi_z^2\langle/\texttt{massitem}\rangle$$

2. $\{\phi_1^2, \cdots, \phi_z^2\} = \bigcup\{\Gamma(\phi_{i_l}^1) | l = 1, \cdots, z\}$

Def. 8 *Let the abstract term τ be a belfunction-valid component on Ω_1. Let Ω_1 and Ω_2 be two compatible frames, and X be a logical variable. If the Prolog predicate* `Propagate`(τ, X) *succeeds, then X is ground to the abstract term τ' that denotes the propagated belfunction-valid component on Ω_2 obtained by Definition 7.*

The predicate defined above can be used to generate an XML document from the first XML document in Example 2 as below.

```
⟨belfunction⟩
  ⟨mass value = “0.2”⟩
    ⟨massitem⟩water⟨/massitem⟩
    ⟨massitem⟩oil⟨/massitem⟩
    ⟨massitem⟩gas⟨/massitem⟩
  ⟨/mass⟩
  ⟨mass value = “0.8”⟩
    ⟨massitem⟩sand⟨/massitem⟩
    ⟨massitem⟩stone⟨/massitem⟩
  ⟨/mass⟩
⟨/belfunction⟩
```

4 Merging Uncertain Information on Subtrees

To merge uncertain components on subtrees, we need to look at **projection** and **extension** operations in DS theory [9].

Def. 9 *Let $V = \{r_1, r_2, \ldots, r_n\}$ be r variables each of which has a set of values or configurations represented by its associated frame of discernment in set $S = \{\Omega_1, \Omega_2, \ldots, \Omega_n\}$. Let $V_p \subseteq V$ and $V_q \subseteq V$ be two subsets of variables where $V_p \subset V_q$, and let $\Omega_p = \otimes_i \Omega_{1i}$ and $\Omega_q = \otimes_j \Omega_{2j}$ be two joint frames for them. Let $Q \subseteq \Omega_q$ be a set of configurations of V_q. Then, the* **projection** *of Q to V_p, denoted by $Q^{\downarrow V_p}$ is a set of configurations for V_p. On the other hand, let H be a subset of Ω_p, then the* **extension** *of P to Ω_q, denoted by $H^{\uparrow V_q}$ is $H \otimes \Omega_{V_q \setminus V_p}$ which is a set of configurations for variable set V_q.*

Let $V_p \subseteq V$ and $V_q \subseteq V$ be two subsets of variables where $\emptyset \neq V_p \subset V_q$. Let m be a mass function on Ω_{V_q} for the joint variable V_q, then the **marginal** of m on Ω_{V_p} for the joint variable V_p, denoted by $m^{\downarrow V_p}$ is a mass function on Ω_{V_p} defined by

$$m^{\downarrow V_p}(H) = \Sigma_{H \subseteq \Omega_{V_p}} \{m(G) | G \subseteq \Omega_{V_q}, G^{\downarrow V_p} = H\}$$

Equally, if m is a mass function on Ω_{V_p} for the joint variable V_p, then the **marginal** of m on Ω_{V_q} for the joint variable V_q, denoted by $m^{\uparrow V_q}$ is a mass function on Ω_{V_q} defined by

$$m^{\uparrow V_q}(G) = \Sigma_{G \subseteq \Omega_{V_q}} \{m(H) | H \subseteq \Omega_{V_p}, H^{\uparrow V_q} = G\}$$

Example 4 *Assume that $\{r_1, r_2, r_3, r_4\}$ are four variables taking values from frames of discernment Ω_i, $i = 1, 2, 3, 4$ respectively, where $\Omega_1 = \{\omega_{11}, \omega_{12}\}$, $\Omega_2 = \{\omega_{21}, \omega_{22}, \omega_{23}\}$, $\Omega_3 = \{\omega_{31}, \omega_{32}, \omega_{33}\}$, and $\Omega_4 = \{\omega_{41}, \omega_{42}, \omega_{43}, \omega_{44}\}$. Let $V_p = \{r_1, r_2\}$ and $V_q = \{r_1, r_2, r_3\}$ be two subset of variables and $Q = \{< \omega_{11}, \omega_{21}, \omega_{31} >, < \omega_{12}, \omega_{23}, \omega_{31} >\}$ be a set of configurations for V_q, then $Q^{\downarrow V_p} = \{< \omega_{11}, \omega_{21} >, < \omega_{12}, \omega_{23} >\}$ is a set of configurations for V_p. On the other hand, given a set of configurations $H = \{< \omega_{11}, \omega_{21} >, < \omega_{12}, \omega_{23} >\}$ for V_p, the extension of it to variable set V_q would be*

$Q' = H^{\uparrow V_q} = \{< \omega_{11}, \omega_{21}, \omega_{31} >, < \omega_{12}, \omega_{23}, \omega_{31} >, < \omega_{11}, \omega_{21}, \omega_{32} >, < \omega_{12}, \omega_{23}, \omega_{32} >, < \omega_{11}, \omega_{21}, \omega_{33} >, < \omega_{12}, \omega_{23}, \omega_{33} >\}$.

Based on the above discussion, we can provide two procedures (similar to Def. 7) that implement both **projection** and **extension** operations and then to define two corresponding predicates for them. Due to the limit of space, we only provide predicate **projection** here and use an example to show the result of using it. Details on projection and extension procedures can be found in [2].

Def. 10 *Let the abstract term τ be a subtree-belfunction-valid component on a subtree with variable set V_q. Let V_p be a subset of V_q and X be a logical variable. If the Prolog query* `Projection`(τ, V_p, X) *succeeds, then X is ground to the abstract term τ' that denotes the propagated subtree-belfunction-valid component on a subtree with variable set V_p.*

Example 5 *Let τ denote the subtree-belfunction-valid uncertain component in Example 3. Using the predicate* `Projection`$(\tau, \{$`deposit`$\}, X)$ *we obtain a new subtree-belfunction-valid component as*

```
⟨deposit⟩
 ⟨belfunction⟩
  ⟨mass value = “0.4”⟩
   ⟨massitem⟩water⟨/massitem⟩
   ⟨massitem⟩oil⟨/massitem⟩
  ⟨/mass⟩
  ⟨mass value = “0.6”⟩
   ⟨massitem⟩gas⟨/massitem⟩
  ⟨/mass⟩
 ⟨/belfunction⟩
⟨/deposit⟩
```

Since there is only one variable to project on when using this predicate, a subtree structure is reduced to a belfunction-valid component on a textentry `deposit`.

5 Conclusion

In this paper, we discussed a method to model and merge mass functions assigned to textentries with different levels of granularity and to subtrees in semi-structured information in the form of XML documents. We use such XML documents to represent structured scientific knowledge (SSK) to provide summaritive or evaluative information, so that users can avoid being overwhelmed by the details implied in the data source. Having information in the form of SSK reports has allowed us to define Prolog predicates that can merge the uncertain information.

```
⟨city⟩
 ⟨Dist⟩
  ⟨Val Prob = “0.7”⟩London⟨/Val⟩
   ⟨outlook⟩
    ⟨Dist type = “mutually − exclusive”⟩
     ⟨Val Prob = “0.1”⟩sunny⟨/Val⟩
     ⟨Val Prob = “0.7”⟩rain⟨/Val⟩
    ⟨/Dist⟩
   ⟨/outlook⟩
  ⟨/Val⟩
  ⟨Val Prob = “0.4”⟩Greater London⟨/Val⟩
   ⟨outlook⟩
    ⟨Dist type = “mutually − exclusive”⟩
     ⟨Val Prob = “0.2”⟩sunny⟨/Val⟩
     ⟨Val Prob = “0.6”⟩rain⟨/Val⟩
    ⟨/Dist⟩
   ⟨/outlook⟩
  ⟨/Val⟩
 ⟨/Dist⟩
⟨/city⟩
```

Figure 1: An XML report using the framework in ProTDB [7].

Because the main focus of the paper is on how to integrate DS theory and its developments into XML structure and how to merge XML documents that involving uncertainties in the format of mass functions, we did not include research results that justify the propagations and combinations of mass functions reported in the paper. These results can be found in relevant publications. Instead, we emphasized on how such information when encoded into XML structure, can be merged and how this procedure can be formally described in logical terminologies and then be executed as Prolog predicates.

In [7], a probabilistic XML model was presented to deal with information with uncertainty that was in the form of probabilities. Using this model, we can construct an XML report as in Figure 1. 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 ⟨outlook⟩). For the second type, two single atoms can be compatible, so the total sum of probabilities can be greater than 1 (as for ⟨city⟩). 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 or without probability), and so on until up to the root of the tree.

The main advantage of this model is its ability to calculate the impact of uncertainty on different levels of an XML document in the form of conditional probabilities. However, it does not merge multiple probabilistic XML on the same issue. On the contrary, our uncertainty XML model focuses on multiple XML datasets and provides a set of means to propagate and merge opinions with uncertainty form different sources. Therefore, our research is complementary to that in [7]. We will discuss how to consider the impact of uncertainty of a parent tag on its child tags in a future paper.

References

[1] J. Cowie and W. Lehnert (1996). Information extraction. *Communications of the ACM*, 39:81–91, 1996.

[2] A. Hunter and W. Liu (2004). Merging uncertain information with semantic heterogeneity in XML. UCL Computer Science Dept Technical Report, 2004.

[3] A. Hunter (2002). Logical fusion rules for merging structured news reports. *Data and Knowledge Engineering*, 42:23–56, 2002.

[4] A. Hunter (2002). Merging structured text using temporal knowledge. *Data and Knowledge Engineering*, 41:29–66, 2002.

[5] J. Lowrance, T. Garvey and T. Strat (1986). A framework for evidential reasoning systems. *Proc. of AAAI'86*, 896-903, 1986.

[6] W. Liu, *et al* (1993). An extended framework for evidential reasoning systems. *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 7(3), 441-457, 1993.

[7] A. Nierman and H. Jagadish (2002). ProTDB: Probabilistic data in XML. In *Proceedings of (VLDB'02)*, LNCS 2590, 646–657. Springer, 2002.

[8] G. Shafer (1976). *A Mathematical Theory of Evidence.* Princeton University Press, 1976.

[9] G. Shafer, P Shenoy, and K Mellouli (1987). Propagating belief functions in qualitative Markov tree. *Int. J. of Approximate Reasoning* Vol 1, 349-400, 1987.