Integration of web sources under uncertainty and dependencies using probabilistic XML

M. Lamine Ba[1], Sebastien Montenez[1], Ruiming Tang[2], and Talel Abdessalem[1]

[1] Institut Mines-Télécom; Télécom-ParisTech; LTCI Paris, France
mouhamadou.ba@telecom-paristech.fr
sebastien.montenez@telecom-paristech.fr
talel.abdessalem@telecom-paristech.fr
[2] National University of Singapore, Singapore
tangruiming@nus.edu.sg

Abstract. We study in this vision paper the problem of integrating several web data sources under uncertainty and dependencies. We present a concrete application with web sources about objects in the maritime domain where uncertainties and dependencies are omnipresent. Uncertainties are mainly caused by imprecise information trackers and imperfect human knowledge. Dependencies come from the recurrent copying relationships occurring among the sources. We answer the issue of data integration in such a setting by reformulating it as the merge of several uncertain versions of the same global XML document. As an initial result, we put forward a probabilistic XML data integration model by getting some intuitions from the versioning model with uncertain data we proposed in [5]. We explain how this model can be used for materializing the integration outcome.

1 Introduction

Uncertain data integration. Data integration with uncertainty, in the form of a probabilistic mediated schema [3, 7, 8] or probabilistic reconciled databases [2, 17], was previously dealt in both relational and XML settings. Probabilistic mappings [7, 8], yielding a probabilistic mediated schema, specify the different possible ways of matching the attributes in multiple relational schemas with respect to their semantics. Query views in [2] through containment constraints are used to define the mappings between a set of uncertain sources and a probabilistic mediated database having a fixed schema. Probabilistic trees in [17] with possibility and probability nodes enable to synchronize several uncertain XML documents by enumerating the different alternatives in data values. Used mappings, query views and reconciliation methods do not care about possible dependencies between sources during the integration process, and thereby they may fail in modeling the set of possibilities and probabilities on the presence of dependencies. One reason is they only assume, in general, that sources describe information about the same real-world objects in an *independent* manner.

Studied problem. We consider in this paper the problem of integrating several web data sources under *uncertainty* and *dependencies*. On the web, many systems collect

and keep up-to-date as much as possible a vast amount of information covering various real-life areas. Uncertainty in web data sources is a well-known issue, on the one hand. On the other hand, the existence of sources that copy (or crawl) data from some others is also a reality. The latter observation translates, as shown in [9, 10, 13], dependencies among the sources in terms of (independent) providers and copiers. Uncertainty and dependencies are two issues particularly true in web sources for objects in the *maritime domain* as we will detail in the next.

A large amount of information related to objects in the maritime domain can be found and extracted from numerous web sources with different nature. Web platforms such as marinetraffic.com, grosstonnage.com and shippingexplorer.net maintain specifications of ships (or boats) and monitor in real-time their locations and routes. Wikipedia, the most popular and successful collaborative content-based web editing platform, contains basic information about some kinds of ships. It harnesses for this the power of the crowd, i.e., its contributors around the world. Last but not least, we have also social networks like Twitter and Flickr that inform their followers (who are interested) about the ship routes with posts and tweets of users. Through the aforementioned source examples, we can easily see that there is enormous knowledge on the web which might be valuable for maritime actors, especially for monitoring the traffic. In order to take advantage of all this knowledge, users within such a context may probably want to be able to transparently query and visualize the integration of information from these sources. Unfortunately, uncertainties and dependencies are omnipresent when talking about web data sources in the maritime domain. Uncertainties are mainly caused by (a) imprecise trackers, also known as automatic identification systems (abbr. AIS) in the maritime domain, that send at constant time intervals ship data like their positions and; (b) unreliable users sharing information about ships in a collaborative or social manner. Dependencies correspond to the copying relationships between sources as we illustrate in Section 2. Consequently, setting up the outcome of the integration of these sources requires to deal with uncertainty and dependencies.

Consider as a running example three sources s_1, s_2 and s_3 sharing a subset of objects in the maritime domain. Let us focus on values they give for the *draft* and the *actual port* for a particular ship named "Costa Serena". The first source s_1 independently reports that the draft of this ship is $8.2m$ and it is currently located at the port of "Hamburg". The second source s_2 which relies on s_1 revises the data it copies by updating the draft and the current location to $8.3m$ and "Marseille", respectively. Finally, the third source s_3 independently sets the draft to $8.3m$ while being not aware about the location. Observe that s_3, taken separately, retains *incomplete* information about the considered ship. Now, assume that one issues a query Q requesting the draft and the current port of this ship based on the three sources. A deterministic data integration system will fail (or eventually it will make an arbitrary choice) by trying to merge the set of answers obtained from sources because there are contradictions resulting from two possible values for the draft and the port name. Eventually, one may prefer to know the set of all possible answers together with their probabilities. Indeed, the user may trust some specific sources and based on this he (or she) may filter the result. Observe that, the user may specify its preferences beforehand with the query. Filtering can also be done according to probabilities. A probabilistic approach for data integration eases this

interaction while enabling to capture *contradictions*, *corroborations* and *incompleteness*. Modeling contradictions and corroborations, and thereby the correct probability values, need to take into account dependencies between the sources. Considering again our example, information about this ship can be obtained independently from s_1 and s_3 whereas requesting some piece of data from s_2 require to use s_1. On the other hand, according to dependencies s_2 disagrees with s_1 while being corroborated by s_3 for the draft. Both scenarios call for being able to maintain the history of the evolution of each piece of information about shared objects during the integration process and according to source dependencies.

Envisioned approach. Source dependencies, in terms of copying relationships, enquire about the history of the evolution of data items about shared objects (copiers may revise collected data as shown in our running example). This setting is very similar to a versioning process on the *web* implying a set of shared objects with uncertainty. Consequently, the problem of integrating web sources under uncertainty and dependencies can be answered by reformulating it as merging several uncertain versions of the same data. We proposed in [5] an uncertain version control model for tree-structured documents with uncertainty that computes the set of *possible versions* together with their *probabilities* by modeling data incompleteness, contradictions and corroborations with respect to the derivation links between data versions. As an enumeration of the set of all possible versions can be unfeasible at a certain limit, our model also includes an efficient compact way to integrate as a whole all these possible versions.

In this vision paper, we present initial ideas towards a probabilistic XML approach for integrating web data sources under uncertainty and (deterministic) dependencies by getting some intuitions from our uncertain versioning model in [5]. As a data model, we choose XML because the tree-like structure of web data can be easily described with this format. We envision a probabilistic integration model able to represent and to assess the amount of uncertainty in both data and sources by modeling possible dependencies between sources. Those dependencies might enable (a) to properly detect data provenance, contradictions, correlations, etc and based on this; (b) to trace the history of the evolution of each piece of information about shared objects. To do so, we associate *event variables* to uncertain sources in order to manage the amount of uncertainty in their own data (its really provided data) and their reliabilities. We adopt a *reconciled* data integration approach which abstracts the result of the integration of several web data sources under uncertainty and dependencies as a set of possible XML documents corresponding to sets of valid events. These sets of events can be used to estimate the probability values of these possible integration results given a probability distribution over the event variables. Concretely, we define the outcome of the integration, i.e., all possible XML integration trees, together with their mapping event sets, as a PrXML$^{\text{fie}}$ probabilistic XML document (see Section 3 about its definition) similarly to [5]: nodes in this special tree are annotated by proportional formulas over event variables which track possible integration results and their probabilities with respect to dependency constraints. Figure 1 is the PrXML$^{\text{fie}}$ probabilistic XML encoding of the result of the integration of sources s_1, s_2 and s_3 of our running example with e_1, e_2 and e_3 their respective associated events.

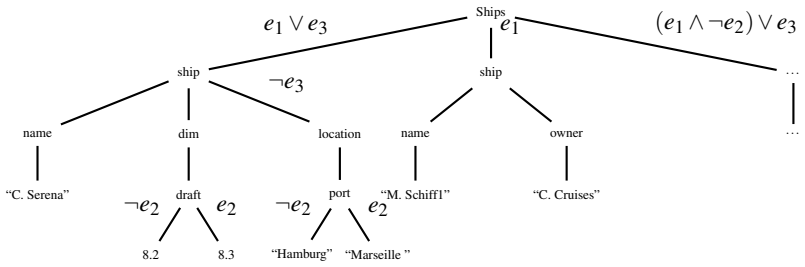


Fig. 1. Probabilistic XML encoding of the integration of shared (uncertain) objects

Outline. The rest of this paper is organized as follows. Section 2 motivates more the problem investigated in this paper by giving concrete examples from the maritime domain. Section 3 revisits some definitions pertaining for the modeling of web semi-structured data, semi-structured uncertain data and multi-version web data with uncertainty. Section 4 presents initial ideas towards a probabilistic XML approach for integrating uncertain tree-structured data sources under dependency constraints. Section 5 concludes the paper and presents some further work.

2 Motivating application

Our motivating application is the integration of web sources providing information about objects (e.g., ships or ports) in the maritime domain. As we show next, web sources in the maritime domain are various, uncertain and dependent. Therefore, users may want to simultaneously query or navigate through all these sources via a unique access point (or global reconciled view); to get answers from its trusted sources and; to know the real provenance of data.

2.1 Numerous web sources

We found that there are a lot of potential web sources by searching information about ships. Figure 2 shows a sample list of those sources[3]. The sources in Figure 2(a) consist of the social networks Flickr and Twitter, and the content-based collaborative platform Wikipedia. In Flickr and Twitter, users share photos about ships and some useful information such as their names and their current locations in the form, for instance, of a tweet. In Wikipedia, users collaborate in order to collect the maximum amount of information about the general description of some particular boats. The sources in

[3] All the screen-shots given in Figure 2 were captured January 8th, 2014 from http://www.flickr.com/search/?q=Costa Serena, `http://en.wikipedia.org/wiki/Costa_Serena`, `http://www.shippingexplorer.net/en/vessels/view/14429-costa-serena`, `http://www.marinetraffic.com/ais/details/ships/247187600`, and `http://www.grosstonnage.com/`.

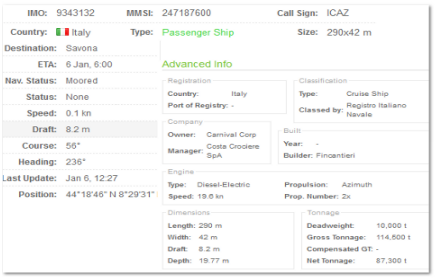
Figure 2(b) are ShippingExplorer, MarineTraffic and GrossTonnage. These sources are mainly dedicated to the monitoring in real-time of the current locations and itineraries of ships even though they also give specifications of these latter. Current locations and itineraries can be known based on data transmitted by AIS systems on ships, for example. There are different levels of *heterogeneity* between the example sources, notably at *schema* level – fortunately, we can observe an homogeneity at object level because sources used in general same unique identifers (e.g., IMO or MMSI) for objects such as ships. In the rest of the paper, we will assume that heterogeneity at schema level is manually resolved with a certain amount of uncertainty.

Flickr

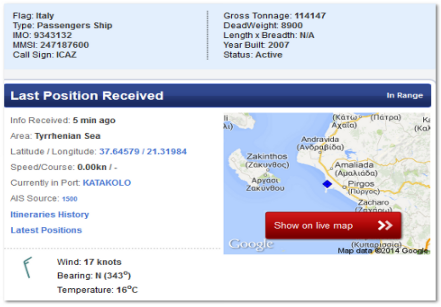
Twitter

Wikipedia

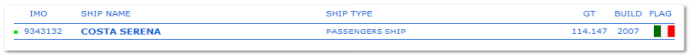
(a) Social networks & collaborative platforms



ShippingExplorer



MarineTraffic



GrossTonnage

(b) Sources monitoring in real-time maritime activities

Fig. 2. Example of web sources about objects in the maritime domain

2.2 Uncertain web data sources

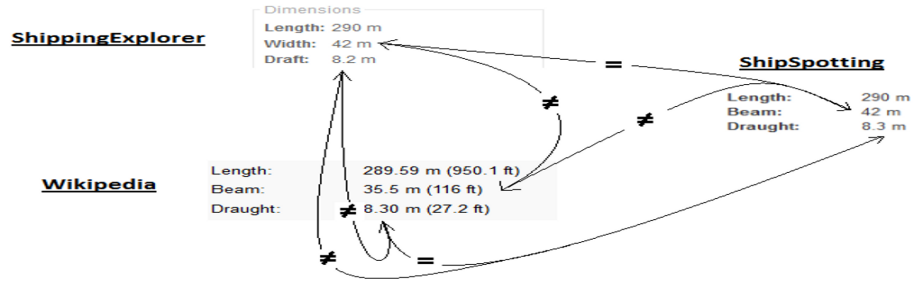
For the maritime domain, web sources are mostly uncertain due to imprecise and imperfect used data extraction methods. The AIS systems are inherently imprecise (e.g., they may transmit incomplete information) whereas human knowledge is imperfect. Other important features in sources revealing the presence of uncertainties are contradictions and incompleteness. Let us analyze the examples in Figures 3(a) and 3(b) which respectively give the company (owner and manager) and the dimensions (length, width and draft) of the same ship from ShippingExplorer, Wikipedia and ShipSpotting. Obviously, we can observe that ShippingExplorer and Wikipedia agree on the owner of this ship (even though Wikipedia seems to be more precise) whereas ShipSpotting gives a different owner. In contrast, all three sources agree on the manager for which the information from ShipSpotting is more complete. As for the dimensions of the ship, it also appears some contradictions between the values given by the sources. For instance, ShippingExplorer and ShipSpotting provide $42m$ for the width while Wikipedia indicates $35.5m$. On another side, Wikipedia and ShipSpotting provide $8.3m$ for the draft whereas ShippingExplorer indicates $8.2m$. For the draft, one may consider that the difference between the indicated values is not very important, and choose the value given by the majority of sources. Unfortunately, determining the correct values for the owner and the width seems to be a more complicated task. A rigorous way to manage contradictions is to keep all the possible values by estimating their *correctness* according to the reliability of the sources.

2.3 Dependent web sources via copying links

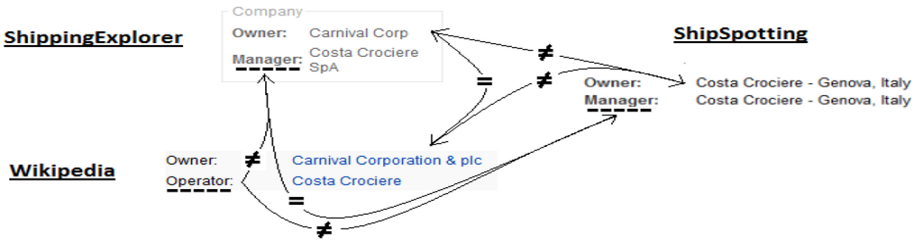
We observed based on the example in Figure 4 that dependent web sources in the maritime domain is a reality. Indeed, there are some sources that collect their information from other ones by copying their data, or by aggregation in the case of several sources simultaneously involved. Such a copying relationship can be explicitly mentioned by the copier, for instance as in Figure 4. The sketched screen-shot was captured January 8th, 2014 from `http://www.shipspotting.com/gallery/photo.php?lid=1825000`. However, in practice the copiers are not all known beforehand. Therefore, sometimes we are constrained to compute these dependencies (see [9, 10] for details about the detection of copying relationships between multiple web sources). Since copiers may revise the collected data based on their own knowledge about the shared real-word objects, having the set of dependencies may help to find the real provenance of each data item and detect more easily contradictions, correlations etc. The detection of copying relationships between a set of web sources is beyond the scope of this paper. We consider in the following that the dependencies are given.

3 Data model

We briefly present in this section some definitions pertaining for the modeling of web semi-structured data, semi-structured uncertain data and multi-version web data with uncertainty. We start by introducing unordered XML trees and a specific model of probabilistic XML trees we use in our integration system.



(a)



(b)

Fig. 3. Uncertain web sources: discrepancies and incompleteness

Fig. 4. Example of dependence between Shipspotting, AirNavShipTrax and GrossTonnage

3.1 Unordered XML and p-Documents based on random events

We model web data as unordered XML trees for convenience of the exposition. The consideration of an order between data items is left to future work.

Unordered XML. Let us consider a finite set L of strings (i.e., labels or text data) and a finite set I of identifiers such that their intersection is empty. We assume also given a labeling function Φ and a identifying function α.

Definition 1. *An unodered XML document is an unordered, unranked, labeled tree* T *of identifiers in* I. *The functions* α *and* Φ *map each node* $x \in$ T *respectively to a unique identifier* $\alpha(x) \in$ I *and to a string* $\Phi(x) \in$ L. *In trees, nodes having at least one child refer to internal nodes, whereas nodes without children are leave with data values.*

For modeling reasons, we use a same node (same label, same identifier) as root of all XML trees referring to the similar object. We omit node identifiers in tree examples for simplicity.

p-Documents based on random events. A probabilistic XML document (abbr. *p-document*) is a compact way of representing a probability distribution over a set of possible unordered XML trees; in the case of interest here, this distribution is finite. A p-document is usually denoted by $\widehat{\mathscr{P}}$ and must be distinguished with a regular XML document as we will see next.

Definition 2. *A probabilistic XML distribution space over a set of uncertain XML trees is a pair* (D,p) *where* D *is a nonempty finite set of possible documents and* $p : D \to (0,1]$ *is a probability function mapping each document* d *in* D *to a rational number, i.e., its probability,* $p(d) \in (0,1]$ *with* $\sum_{d \in D} p(d) = 1$.

We draw our attention on the most expressive and succinct family of p-documents from [11], namely PrXML$^{\text{fie}}$ model with fie standing for formula of independent events. For a complete insight about existing probabilistic XML models, see [1, 12]. Let B be a set of independent random Boolean variables (abbr. *event variables*) $b_1 \dots b_m$. The truth of each event variable b_i is given by its probability value $Pr(b_i)$ of being valid. We revisit below the *syntax* and the *semantics* of the encoding of a probability distribution using a p-document of the PrXML$^{\text{fie}}$ family.

Definition 3. *A p-document* $\widehat{\mathscr{P}}$ *based on independent random variables is an unordered, unranked, and labeled XML tree in which (i) the root is always certain and; (ii) all other node* x *may be annotated with a propositional formula* $\text{fie}(x)$ *of events* $b_1 \dots b_m$.

A proportional formula represents and estimates the amount of uncertainty in its attached node. Some distinct formulas may be correlated by sharing common events. At last, the number of event variables in the formulas is not necessarily the same. The set of all possible XML trees obtainable from $\widehat{\mathscr{P}}$ defines its *possible worlds*. Those possible worlds are produced in function of the different possible ways of valuating the event variables. A valuation ν of variables $b_1 \dots b_m$ is a mapping of each b_i to true or false. This valuation generates, when it is evaluated over $\widehat{\mathscr{P}}$, one particular XML tree $\nu(\widehat{\mathscr{P}})$ consisting only of nodes from $\widehat{\mathscr{P}}$ whose formulas are valuated at true with ν. We denote the possible worlds of $\widehat{\mathscr{P}}$ by $\mathscr{D}(\widehat{\mathscr{P}})$. Let $\hbar[\nu]$ be the set of all possible valuations over variables $b_1 \dots b_m$. The probability of a possible world $d \in \mathscr{D}(\widehat{\mathscr{P}})$ is given hereafter.

$$Pr(d \mid d \in \mathscr{D}(\widehat{\mathscr{P}})) = \sum_{\substack{v' \in \hbar(v) \\ v'(\widehat{\mathscr{P}})=d}} Pr\Big(\bigwedge_{\substack{b_i \in \mathrm{B} \\ v'(b_i)=\mathsf{true}}} b_i \wedge \bigwedge_{\substack{b_j \in \mathrm{B} \\ v'(b_j)=\mathsf{false}}} \neg b_j \Big). \quad (1)$$

Definition 4. *The semantics $[\![\widehat{\mathscr{P}}]\!]$ of a p-document $\widehat{\mathscr{P}}$ in the probabilistic XML model based only on formula of independent random variables is the distribution (D,p) defined in such that (a) $D = \mathscr{D}(\widehat{\mathscr{P}})$ and (b) for all $d \in D$, $p(d) = Pr(d|d \in \mathscr{D}(\widehat{\mathscr{P}}))$.*

3.2 Semi-structured multi-version data with uncertainty

In a XML setting, a semi-structured multi-version data with uncertainty (typically, shared web data in our context) is defined in [5] as evolving through uncertain updates and leading to uncertain versions. We summarize here this model which describes such a multi-version data with the help of two components: the *derivation graph* between the data versions (or version space[4]) and a *probability distribution* over a set of possible XML tree versions. For more details about the model and its original context of use, we refer to [4, 5].

We suppose a set of complex event variables $e_1 \dots e_n$, each representing a conjunction of atomic event variables $b_1 \dots b_m$. Considered events model the different uncertain states of the multi-version document. As a result, an event has also contextual information about a given version, in particular the edit script δ_i (i.e., a sequence of insertions and deletions) leading to it.

Definition 5. *A multi-version XML document with uncertainty is a pair (G, ω) where G is a directed acyclic graph (DAG) of events $\{e_0\} \cup \{e_1 \dots e_n\}$ representing the derivation graph of the tree versions, and ω is a probability distribution over the set of possible document versions.*

The special event e_0 is the root of G and maps to the initial state of the multi-version document. A version is an unordered XML tree mapping to a set of events in G whose edit scripts together made this version happen. Given the infinite set $\mathbb{D}$ of all unordered XML trees, we have $\omega : 2^{\{e_1 \cdots e_n\}} \to \mathbb{D}$ with (a) $\omega(\{\})$ a root-only XML tree and; (b) for all i, for all $\mathrm{F} \subseteq 2^{\{e_1 \cdots e_n\} \setminus \{e_i\}}$, $\omega(\{e_i\} \cup \mathrm{F}) = [\omega(\mathrm{F})]^{\delta_i}$ ($[\omega(\mathrm{F})]^{\delta_i}$ results from applying δ_i on $[\omega(\mathrm{F})]$). This mapping corresponds to a probability distribution, compactly encoded in [5] as $\widehat{\mathscr{P}}$, over a set of possible trees versions.

Definition 6. *A compact representation system of a multi-version XML with uncertainty is a pair $(\mathrm{G}, \widehat{\mathscr{P}})$ where (a) G is the DAG of events $e_0 \dots e_n$ and; (b) $\widehat{\mathscr{P}}$ is a $\mathrm{PrXML}^{\mathrm{fie}}$ p-document with random variables $b_1 \dots b_m$ encoding compactly all the possible tree versions and their mapping event sets.*

In this compact representation, given a node $x \in \widehat{\mathscr{P}}$ and its associated formula $\mathrm{fie}(x)$, *corroborations* and *contradictions* are modeled as follows for all event e_i.

[4] The version space describes the history of the editing process by maintaining necessary information about the different versions and their dependencies.

- If $e_i \models \text{fie}(x)$, then there is a corroboration of the presence (or validity) of x at this event e_i.
- If $e_i \not\models \text{fie}(x)$, then this means that event e_i contradicts the existence of x (or invalid x).

4 Heterogeneous web data integration using probabilistic XML

We elaborate in this section our probabilistic XML model dealing with the integration of web data sources under uncertainty and dependencies. We first present some challenges underlying the set up of the intended model. We then put forward a model and explain how it can be used for materializing the integration in the scenario where the dependencies between sources are *deterministic*.

Consider a set S of n web sources $S_1, \ldots, S_n$ under uncertainty and dependencies. For the sake of simplicity, we assume that (a) all of these sources maintain information about real-world objects in the same domain; (b) the objects are distinguished each other by a unique identifier and; (c) each local source S_i provides data about its subset of tracked objects in the form of a global unordered XML tree that we denote also S_i. Moreover, we assume that, first, a dependency relationship involving two sources, if it occurs, is directed and there is no cycles; second, each dependency relationship is deterministic, that is, it is known with certainty.

4.1 Main challenges

We base the design of our intended data integration model for uncertain web data sources with dependencies on three main requirements.

1. At first, since we have uncertainties on the sources and on the data, we need a way to represent and to evaluate these uncertainties during the integration. The used model must be flexible enough to enable (a) correlating the uncertainty about the sources and the provided information; (b) tracking the provenance of each data item. Obviously, for instance, one may trust a given source but considers that its data are invalid. As a result, the model must enable *explanation* and *understanding* of obtained probability values.
2. Second, we have to introduce a technique for finding and representing the dependency relationships between sources; in this work we consider that these dependencies are known beforehand.
3. Finally, the integration approach, formalizing the result of the integration and its semantics mapping with the local sources (or partial views of these latter), must be defined by focusing especially on the uncertain nature of our setting and the dependencies (a given source first copies other ones, and then it may revise the copied data with new knowledge). Therefore, the used model must enable the modeling of contradictions and corroborations in the integration outcome.

4.2 Probabilistic XML integration framework

Here we give a first attempt for formalizing our intended probabilistic XML integration framework. We first go further on each requirement aforementioned by translating the version control model with uncertainty, we designed in [5] (see Section 3 for a summary), in our data integration setting. Then, we formalize the proposed model.

Dealing with uncertainties. Similarly to [5], we use random event variables in order to deal with uncertainties. Consider again B as a set of independent random Boolean variables $b_1 \dots b_m$ and their probability values $Pr(b_1) \dots Pr(b_m)$ of being true as well. We restrict B to two types of disjoint sets of variables which we denote $\mathrm{B_r}$ and $\mathrm{B_s}$. As in [5], we use variables in $\mathrm{B_r}$ to manage the uncertainty in the content really provided by each source: the data it does not copy from others sources (its contribution in a certain sense). Variables in $\mathrm{B_s}$ are used to model the trust one can have on sources. Given a source S_i, we refer to the uncertainty on its content and its reliability level with $b_{r,\,i}$ and $b_{s,\,i}$ respectively. We consider now the set of events $e_1, \dots, e_n$. In order to represent and evaluate the overall amount of uncertainty in each source S_i from S, an event $e_i = b_{r,\,i} \wedge b_{s,\,i}$ with $b_{r,\,i} \in \mathrm{B_r}$ and $b_{s,\,i} \in \mathrm{B_s}$ is associated to it. Intuitively, e_i is true when it produces a correct content on a reliable source. The probability associated to it is obtained by computing the probability of the corresponding conjunction. It estimates numerically its correctness.

Representing dependencies between sources. Detecting dependencies between a set of web sources in the same domain has been mainly dealt in [9, 10]. As shown in these papers, such dependencies follow a DAG structure G. More precisely, we define a DAG with events (representing sources) as nodes.

Definition 7. *Given the set of events $e_1 \dots e_n$ associated to $s_1 \dots s_n$, formally we set $G = (V_G, E_G)$ where V_G is the set $\{e_0\} \cup \{e_1, \dots e_n\}$ representing the nodes of the DAG; $E_G \subseteq V_G \times V_G$ is the set of edges of the DAG tracking (implicitly) the dependencies between sources.*

Conventionally, we introduce the special event e_0 as the root of the DAG. The events associated to independent sources are linked to e_0.

Data integration approach. As some previous work on uncertain tree-structured data integration, such as the paper of Van Keulen et al. [17], we build our system on a probabilistic model in a *reconciled* fashion. To do so, we introduce the notion of a *probabilistic XML global view* (*PrGView*) $\mathbb{M}$ which is a set of *possible* integrated XML documents $m_1, \dots, m_k$ with probabilities $Pr(m_1), \dots, Pr(m_k)$ attached to them. An integrated XML document m_i is defined as a deterministic XML document in $\mathbb{D}$ resulting from the integration of changes from multiple XML documents. There is not only one valid integrated XML document in the presence of uncertainty, but *several* describing, first, the views on the trust one may have on the given sources and their data; second, the different way to deal with contradictions and incompleteness of the data. We show later that *PrGView* enables to capture all such possible results of an integration.

Definition 8. *Let $s_1, \ldots, s_n$ be a set of uncertain sources with dependencies in* G*. A PrGView* $\mathbb{M}$ *of the integration process over* $\{s_1, \ldots, s_n\}$ *is a set* $\{(m_1, Pr(m_1)), \ldots, (m_k, Pr(m_k))\}$ *where (i) for each* $1 \leq i \leq k$*,* m_i *is a possible integrated XML document for* $s_1, \ldots, s_n$ *constrained by dependencies; (ii)* $0 < Pr(m_i) \leq k$ *with* $\sum_{i=1}^{k} Pr(m_i) = 1$*.*

We need to define how we obtain $\mathbb{M}$ based on the set of input sources. We will focus more on the representation of the set of possible worlds than on their probabilities. We start by defining the *contribution* of a given source within our integration setting.

Definition 9. *Let $s_1, \ldots, s_n$ be a set of uncertain sources with dependencies in* G*. The contribution of any given source s_i, with respect to* G*, corresponds to the real content provided by this source. We represent the contribution of s_i, that we denote δ_i, in the form of a sequence of edit operations over some initial data.*

Given any s_i, let us denote by $S^{(e_i)}$ the set of sources on which s_i depends according to G, that is, for each $1 \leq l \leq n$ with $i \neq l$, $s_l \in S^{(e_i)}$ if the relation (e_l, e_i) exists in G. Algorithm 1 sketches a process for computing the contribution of s_i in function of their dependent sources and their documents; `DIFF` is a *differencing function* (see [6, 14, 15] for more details) that gives the difference between two XML documents. Its output is a sequence of edit operations over XML nodes.

Input: Set $S^{(e_i)}$ of sources on which s_i depends
Output: Compute the contribution δ_i of the source s_i
1 Set $s_0 \leftarrow$ root-only XML tree in $\mathbb{D}$;
2 **if** $S^{(e_i)}$ *is an empty set* **then**
3 | Set $\delta_i \leftarrow \texttt{DIFF}(s_0, s_i)$;
4 **else**
5 | **foreach** *source* s_l *in* $S^{(e_i)}$ **do**
6 | | Set each $\delta_{l,i} \leftarrow \texttt{DIFF}(s_l, s_i)$;
7 | Insert all insertions shared by $\delta_{l,i}$'s in δ_i;
8 | Insert all deletions in each $\delta_{l,i}$ in δ_i;
9 **return** (δ_i);

Algorithm 1: Computation of the contribution of a source

In addition to variables, we attach the contributions $\delta_1, \ldots, \delta_n$ of the sources $s_1, \ldots, s_n$ to their associated events $e_1 \ldots e_n$, respectively. In this setting, the events are enough to describe the sources because they contain the information about both the amount of uncertainty and the data of the considered sources. In the following, we will refer to the sources by their associated events. We construct a mapping ω between possible integrated XML documents in $\mathbb{M}$ and the sources $s_1, \ldots, s_n$ by adopting the model in [5] as follows. Let $\mathbb{D}' \subseteq \mathbb{D}$ such that $\mathbb{D}'$ includes $\{m_1, \ldots, m_k\}$ and the root-only tree document.

Definition 10. *Given a set of uncertain sources $s_1, \ldots, s_n$ with dependencies in* G*. Let $\mathbb{M}$ be the PrGView of the integration of these sources. We define the mapping $\omega : \mathbb{D}' \rightarrow 2^{\{e_1, \ldots, e_n\}}$ as specifying the possible integrated XML documents in PrGView in terms of data contained in the sources such that $\omega(\{\})$ corresponds to the root-only document*

in $\mathbb{D}'$ and; for each $F \subseteq 2^{\{e_1,\ldots,e_n\}\setminus\{e_i\}}$, $\omega(F \cup \{e_i\}) = [\omega(F)]^{\delta_i}$. Let us assume that $m_k = \omega(F)$ for a fixed $1 \leq k \leq m$. The probability of m_k is estimated as follows.

$$Pr(m_k) = \sum_{\substack{F \subseteq \{e_1,\ldots,e_n\} \\ \omega(F)=m_k}} \prod_{\substack{1 \leq i \leq n \\ e_i \in F}} Pr(e_i) \times \prod_{\substack{1 \leq i \leq n \\ e_i \notin F}} 1 - Pr(e_i). \quad (2)$$

According to [5] such a ω mapping, corresponding to the construction of the $\mathbb{M}$, can be encoded efficiently as a PrXML$^{\text{fie}}$ p-document $\widehat{\mathscr{P}}$.

Definition 11. *Given a set of uncertain sources $s_1,\ldots,s_n$ with dependencies in* G. *Following the encoding proposed in [5], we define the* PrXML$^{\text{fie}}$ *p-document $\widehat{\mathscr{P}}$ as an efficient representation of the PrGView M resulting of the integration of $s_1,\ldots,s_n$ with respect to* G.

Based on the definitions given above, we formalize our probabilistic data integration framework as follows.

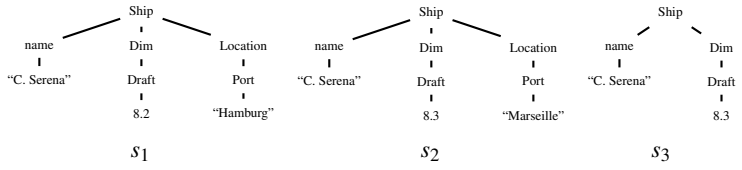
Our model. Let $s_1,\ldots,s_n$ be a set of uncertain sources with dependencies described by a DAG G of events $e_0,\ldots,e_n$. We abstract a probabilistic XML integration model over these sources with the help of a triple $(\mathrm{G},\mathbb{M},\omega)$ where (i) G is a DAG of $\{e_0\} \cup \{e_1,\ldots,e_n\}$ in which each node e_i, for $1 \leq i \leq n$, is associated to a source s_i in order to manage its overall amount of uncertainty and its contribution; (ii) $\mathbb{M}$ is the *PrGView* of the integration of $s_1,\ldots,s_n$ and; (iii) ω is a mapping between the set of possible integrated documents in $\mathbb{M}$ and the sources $s_1,\ldots,s_n$ through their associated events $e_1,\ldots,e_n$.

We define a compact representation of such a probabilistic XML integration model as a pair $(\mathrm{G},\widehat{\mathscr{P}})$ where (a) G remains the same DAG of events $e_0,\ldots,e_n$ and; (b)$\widehat{\mathscr{P}}$ is the PrXML$^{\text{fie}}$ encoding based on events $e_1,\ldots,e_n$ and contributions $\delta_i,\ldots,\delta_n$ of the set of possible integrated XML documents in $\mathbb{M}$.

Example 1. Figure 5 illustrates our integration model with our running example. Figure 5(a) shows the XML corpus of the three sources s_1, s_2 and s_3 where s_2 is a copier of s_1. Figure 5(b) gives the DAG modeling the dependencies between the sources and their contributions δ_1 δ_2, δ_3 which can be estimated with Algorithm 1. Figure 5(c) shows two examples of possible integrated XML documents by reasoning on the validity or not of each event. The first integrated XML document is obtained by only considering as valid e_1 and e_3, thus corresponds, for instance, to the case where a user requests to integrate data from only the two independent sources s_1 and s_3. The integrated version is obtained by evaluating δ_1 and δ_3 on $\omega(\{\})$ and $\omega(\{e_1\})$, sucessively. We can generate all the possible XML integrated documents of the three given uncertain XML corpus under dependencies by following the same process.

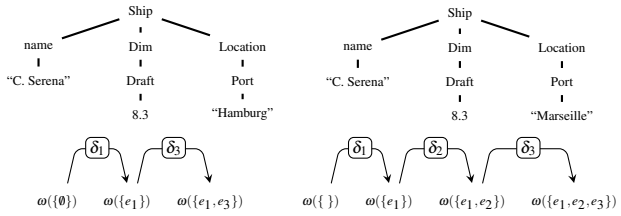
Reduce uncertainties by crowd-sourcing. A probabilistic data integration approach, even in the scenario where the dependencies between the sources are known, only models or resolves a portion of the overall uncertainty in the data and the sources. On the

one hand, the process is itself imprecise. On the other hand, usually there is not enough knowledge about the modeled domain, the semantics of the integrated data etc. As studied in [16], knowledge rules and user feedback can help to resolve the uncertainties in some part of the integration result by refining the set of possible worlds. Crowdsourcing is a reliable way for obtaining additional knowledge, for instance opinions from maritime experts regarding our application domain.



(a) XML corpus of three sources with s_2 depending on s_1

(b) DAG of dependencies and contributions



(c) Example of possible XML integrated documents

Fig. 5. Probabilistic XML integration over XML corpus with uncertainty and dependencies

5 Conclusion and Further work

We have presented initial directions towards a probabilistic XML approach for integrating web sources under uncertainty and dependencies. We first provided a concrete application of such a model. Then, we set up a first abstraction of our integration model by translating the problem in an uncertain version control setting.

Further work could explore the effect of uncertain dependencies in the modeling of the set of possible worlds. It could be also of interest to investigate the definition of the probabilistic XML global view in terms of query views over the sources.

Acknowledgements. We are graceful to Pierre Senellart and Stephane Bressan for their precious remarks and suggestions. This work was partially funded by the NORMATIS project, and the French government under the STIC-Asia program, CCIPX project.

References

1. S. Abiteboul, B. Kimelfeld, Y. Sagiv, and P. Senellart. On the expressiveness of probabilistic XML models. *VLDB Journal*, Oct 2009.
2. P. Agrawal, A. D. Sarma, J. Ullman, and J. Widom. Foundations of uncertain-data integration. *Proc. VLDB Endow.*, Sept 2010.
3. N. Ayat, H. Afsarmanesh, R. Akbarinia, and P. Valduriez. An uncertain data integration system. In R. Meersman, H. Panetto, T. Dillon, S. Rinderle-Ma, P. Dadam, X. Zhou, S. Pearson, A. Ferscha, S. Bergamaschi, and I. Cruz, editors, *On the Move to Meaningful Internet Systems: OTM 2012*. Springer Berlin Heidelberg, 2012.
4. M. L. Ba, T. Abdessalem, and P. Senellart. Merging uncertain multi-version XML documents. In *Proc. DChanges*, Florence, Italy, Sept. 2013.
5. M. L. Ba, T. Abdessalem, and P. Senellart. Uncertain version control in open collaborative editing of tree-structured documents. In *Proc. DocEng*, 2013.
6. G. Cobena, T. Abdessalem, and Y. Hinnach. A comparative study for XML change detection. In *BDA*, 2002.
7. A. Das Sarma, X. Dong, and A. Halevy. Bootstrapping pay-as-you-go data integration systems. In *Proc. SIGMOD*, 2008.
8. X. Dong, A. Y. Halevy, and C. Yu. Data integration with uncertainty. In *Proc. VLDB*, 2007.
9. X. L. Dong, L. Berti-Equille, Y. Hu, and D. Srivastava. Global detection of complex copying relationships between sources. *Proc. VLDB Endow.*, Sept 2010.
10. X. L. Dong, L. Berti-Equille, and D. Srivastava. Integrating conflicting data: the role of source dependence. *Proc. VLDB Endow.*, Aug 2009.
11. E. Kharlamov, W. Nutt, and P. Senellart. Updating probabilistic xml. In *Proc. EDBT/ICDT Workshops*, 2010.
12. B. Kimelfeld and P. Senellart. Probabilistic XML: Models and complexity. In Z. Ma and L. Yan, editors, *Advances in Probabilistic Databases for Uncertain Information Management*. Springer-Verlag, 2013.
13. X. Li, X. L. Dong, K. Lyons, W. Meng, and D. Srivastava. Truth finding on the deep Web: is the problem solved? In *Proc. VLDB*, Sept 2013.
14. T. Lindholm, J. Kangasharju, and S. Tarkoma. Fast and simple XML tree differencing by sequence alignment. In *Proc. DocEng*, 2006.
15. L. Peters. Change detection in XML trees: a survey. In *TSIT Conference*, 2005.
16. M. van Keulen and A. de Keijzer. Qualitative effects of knowledge rules and user feedback in probabilistic data integration. *VLDB Journal*, 18, 2009.
17. M. van Keulen, A. de Keijzer, and W. Alink. A probabilistic XML approach to data integration. In *Proc. ICDE*, 2005.