International Journal of Software Engineering
and Knowledge Engineering
Vol. 24, No. 4 (2014) 635–652
© World Scientific Publishing Company
DOI: 10.1142/S0218194014500247

An Iterative Approach to Managing Uncertain Mappings in Dataspace Support Platforms

Nathalie Cindy Kuicheu*, Ning Wang†,
Gile Narcisse Fanzou Tchuissang‡ and De Xu§

School of Computer and Information Technology
Beijing Jiaotong University, 3 Shangyuancun Xizhimenwai
100044 Beijing, P. R. China
**nathkuicheu@yahoo.fr*
†nwang@bjtu.edu.cn
‡fanzounar2002@yahoo.fr
§dxu@bjtu.edu.cn

Guojun Dai

Computer School, Hangzhou Dianzi University
Hangzhou 310018, P. R. China
daigj@hdu.edu.cn

François Siewe

Software Technology Research Laboratory
De Montfort University, The Gateway, Leicester LE1 9BH, UK
fsiewe@dmu.ac.uk

Received 27 October 2013
Revised 12 January 2014
Accepted 27 February 2014

A DataSpace Support Platform (DSSP) is a self-sustained and self-managed system which needs to support uncertainty among its mediated schemas and its schema mappings. Some approaches for managing such uncertainty by assigning probabilities and reliability degrees to schema mappings have been proposed. Unfortunately, the number of mappings self-generated by a DSSP is usually too large and among those possible mappings, some might be totally correct and others partially correct. Therefore, providing probabilities or reliability degrees to the mappings is necessary but not sufficient to resolve uncertainty among them. This paper proposes a stepper-based approach called pos-mapping to managing reliable mappings using possibility theory. Instead of choosing a threshold for managing the reliable mappings, pos-mapping approach orders and divides the set of reliable mappings into subsets of possibility distributions and assigns to each of these subsets a recursive possibility degree function. The recursiveness of the possibility degree function leads to an incremental management of the possibility

† Corresponding author.

distributions. Experimental results show that our system is more efficient than the existing systems and the accuracy of the results increases with the number of reliable schemas in the DSSP.

Keywords: Schema mapping; reliability degrees; possibility theory; dataspace.

1. Introduction

Dataspace [1, 2] can be thought of as a virtual space where many data sources are managed regardless of their structures and locations; this, with the main goal of providing basic functionalities such as information retrieval or keyword search over all its sources without any domain expert assistance. Semantic mappings should then be created automatically; this leads to multiple possible semantic mappings. Among them, some might be correct and several others partially correct. Therefore, a first type of uncertainty arose between semantic mappings. Probabilistic schema mappings (pmapping)[3] and reliable mediated schema and mapping (rMedMap)[4] have been proposed to handle uncertainty between the multiple possible mappings. Pmapping is a probabilistic model which attaches probabilities to the semantic mappings. Likewise, rMedMap assigns reliability degrees to the semantic mappings. In both pmapping and rMedMap, the system needs to choose a threshold.

Considering the usually large number of semantic mappings, choosing a threshold may become an uncertain task and may lead to information loss and so create a second type of uncertainty among semantic mappings. To address this second type of uncertainty among the semantic mappings, we propose to "fuzzify" reliable mappings using possibility theory [5, 6]. Possibility theory is an effective uncertainty theory devoted to the handling of incomplete information. It is similar to probability theory because it is based on set-functions. It differs from probability by the use of a pair of dual set functions called possibility and necessity measures instead of only one [5]. Our approach can handle uncertainty among large number of semantic mappings without the need of choosing a threshold. Indeed, we order and divide the set of reliable mappings into subsets of possibility distributions and we assign to each of the subset a recursive possibility degree function of being the most reliable mapping between a source schema and a target reliable mediated schema.

The contributions of this paper are summarized as follows:

(1) To the best of our knowledge, this is the first work that uses the possibility theory to manage uncertainty in DSSP.
(2) We propose an iterative method for computing possibility degrees whereby the possibility degrees of the current set of possibility distributions is a function of the possibility degrees of the previous set of possibility distributions.
(3) We implement the proposed method, and experimental results show that our system is more efficient than the existing systems and the accuracy of the results increases with the number of reliable schemas in the DSSP.

The rest of this paper is organized as follows. The rMedMap method is summarized in Sec. 2. Section 3 presents the pos-mapping method. Section 4 discusses a comparative evaluation of pos-mapping with the existing rMedMap and pmapping methods. Section 5 introduces related works and discussion. Section 6 concludes this paper and highlights some future works.

2. Background

Previously, we proposed in [4] a reliability degree function which measures how tightly a semantic mapping is related to a given source schema. We summarize below the equations used in [4] to compute reliability degree. Considering an instance $\mathcal{T}$ of the set of possible mediated schemas and a given source schema $\mathcal{S}_i$, the aim is to find out whether $\mathcal{T}$ is reliable with respect to $\mathcal{S}_i$ and assign to $\mathcal{T}$ a reliability degree with respect to $\mathcal{S}_i$. To compute this reliability degree, we first check if, in a structural view point, $\mathcal{T}$ is reliable with respect to $\mathcal{S}_i$ and we call it the *structural reliability degree*. $\mathcal{T}$ is said to be structurally reliable with respect to $\mathcal{S}_i$ if the root node of $\mathcal{T}$ is structurally more general or equivalent to the root node of $\mathcal{S}_i$ and if the structural reliability degree of $\mathcal{T}$ with respect to $\mathcal{S}_i$ is greater than a certain threshold α [4]. The structural reliability degree is computed as the ratio between the numbers of sub-nodes of $\mathcal{T}$ that are structurally equivalent to a sub-node of $\mathcal{S}_i$ and the number of sub-nodes of $\mathcal{S}_i$. The structural reliability degree of $\mathcal{T}$ with respect to $\mathcal{S}_i$ noted $d^{\mathit{Struct}}_{\mathcal{T}/\mathcal{S}_i}$, is given by Eq. (1):

$$d^{\mathit{Struct}}_{\mathcal{T}/\mathcal{S}_i} = \frac{|\mathcal{T} \cap \mathcal{S}_i|}{|\mathcal{S}_i|}. \tag{1}$$

We compute the degree of reliability of $\mathcal{T}$ with respect to $\mathcal{S}_i$ noted $d_{\mathcal{T}/\mathcal{S}_i}$ using the following Eq. (2).

$$d_{\mathcal{T}/\mathcal{S}_i} = \frac{\sum_{e \in \mathcal{T} \cap \mathcal{S}_i} d(e)p(e)}{\sum_{e \in \mathcal{S}_i} d(e)p(e)}, \tag{2}$$

where $d(e)$ is the similarity value between elements of the group to which e belongs; and $p(e)$ is the probability of encountering an instance of the element e in a mediated schema. $p(e)$ is then the ratio between the number of mediated schemas containing e divided by the total number of mediated schemas. Therefore, $\mathcal{T}$ is said to be reliable with $\mathcal{S}_i$ if:

(a) From Eq. (1), $d^{\mathit{Struct}}_{\mathcal{T}/\mathcal{S}_i} \geq \alpha$
(b) From Eq. (2), $d_{\mathcal{T}/\mathcal{S}_i} \geq \beta$.

Finally, the degree of reliability of a given mediated schema $\mathcal{T}$ with respect to the set of data sources $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_n$ will be the average of the degrees of reliability of $\mathcal{T}$ with respect to each of the sources $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_n$ and we write:

$$d_{\mathcal{T}/\mathcal{S}_1,\mathcal{S}_2,\ldots,\mathcal{S}_n} = \frac{\sum_{i=1}^{n} d_{\mathcal{T}/\mathcal{S}_i}}{n}. \tag{3}$$

Thereby defined, considering a given source $\mathcal{S}_i$ and 2 mediated schemas $\mathcal{T}_j$ and $\mathcal{T}_k$ such that $\mathcal{T}_j = \mathcal{T}_k$, then $\mathcal{T}_j \cap \mathcal{S}_i = \mathcal{T}_k \cap \mathcal{S}_i$, i.e. $d_{\mathcal{T}_j/\mathcal{S}_i} = d_{\mathcal{T}_k/\mathcal{S}_i}$. Besides, $\sum_{e_i \in \mathcal{S}_i} d(e)p(e) \neq 0$ because $d(e) \neq 0$ is a function of $sim(e_i, e_j) \neq 0$ and $p(e_i) \neq 0$ since only the highly similar elements are considered when constructing the mediated schema and reliability degrees are computed when the structural reliability degree is greater than a certain threshold. Therefore, reliability degree function thus defined is well defined.

3. Handling Uncertainty between Reliable Mappings

This section presents the principles of the possibility theory based method called *pos-mapping* for managing uncertainty between reliable mappings. We begin with a brief introduction of the possibility theory.

3.1. *Possibility Theory*

Possibility theory is an effective uncertainty theory devoted to the handling of incomplete information. It is similar to probability theory [7] because it is based on set-functions. It differs from probability by the use of a pair of dual set functions called *possibility* and *necessity* measures instead of only one [5]. Necessity and possibility measures compute respectively the minimum and the maximum level of certainty or degree of reliability in a given domain of application. In other words, possibility degree of a disjunction of events is the maximum of the possibility degrees of individual events. In the contrary, the necessity degree of a conjunction of events is the minimum of necessity degrees of the individual events [6].

The use of maximum and minimum operations, along with the complement to 1, is in agreement with the requirement of computational simplicity and with the rather rough and qualitative nature of the uncertainty that can be expressed in many real world applications [8]. It should be noted that in possibility theory, the modeling of uncertainty may remain qualitative. Indeed, we could use a finite completely ordered chain of level of reliability denoted here by λ ranging between 0 and 1, i.e. $\lambda_1 = 0 < \lambda_2 < \cdots < \lambda_n = 1$ instead of the whole interval $[0, 1]$, with $min(\lambda_i, \lambda_k) = \lambda_i$ and $max(\lambda_i, \lambda_k) = \lambda_k$, for $i \leq k$.

3.2. *Pos-mapping principle*

Let us first recall that the aim to build a DSSP is to provide users with basic functionalities such as information retrieval or keyword search. Therefore, the result produced by the system should consider all the information available in the sources connected to the DSSP. Previously, we introduced rMedMap [4], a reliability based method which enables the system to exploit as much as possible information available in the sources compared to pmapping [3], a probability based method which considers less information available. In this paper, the proposed pos-mapping method exploits all the information available in the sources connected to the DSSP. In fact,

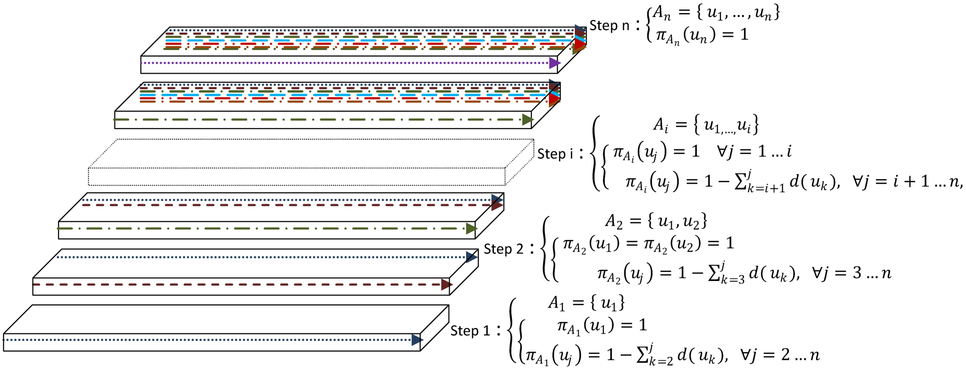
Step n : $\begin{cases} A_n = \{u_1, \ldots, u_n\} \\ \pi_{A_n}(u_n) = 1 \end{cases}$
Step i : $\begin{cases} A_i = \{u_{1,\ldots,}u_i\} \\ \begin{cases} \pi_{A_i}(u_j) = 1 \quad \forall j = 1 \ldots i \\ \pi_{A_i}(u_j) = 1 - \sum_{k=i+1}^{j} d(u_k), \quad \forall j = i+1 \ldots n, \end{cases} \end{cases}$
Step 2 : $\begin{cases} A_2 = \{u_1, u_2\} \\ \begin{cases} \pi_{A_2}(u_1) = \pi_{A_2}(u_2) = 1 \\ \pi_{A_2}(u_j) = 1 - \sum_{k=3}^{j} d(u_k), \quad \forall j = 3 \ldots n \end{cases} \end{cases}$
Step 1 : $\begin{cases} A_1 = \{u_1\} \\ \begin{cases} \pi_{A_1}(u_1) = 1 \\ \pi_{A_1}(u_j) = 1 - \sum_{k=2}^{j} d(u_k), \quad \forall j = 2 \ldots n \end{cases} \end{cases}$

Fig. 1. Pos-mapping principle.

pos-mapping is a stepper-based method which provides the result of the current set of information available using the result of the latter set of information available. Figure 1 illustrates the pos-mapping principle. At Step 1 for example, the system deals only with a subset A_1 and its corresponding characteristic function π_{A_1}. The subset A_1 is further expanded in step 2 to construct the subset A_2 and its corresponding characteristic function π_{A_2} which also carries the information of the latter characteristic function π_{A_1}, and so on.

3.3. *Running example*

The possible application domains of dataspace includes Personal Information Management, Web-Scale Information Management and Medical Information Management [2]. In an applicative point of view, our objective during our research is to construct a dataspace for medical information management; especially for African Traditional Medicine information management. Figure 2 shows an example of two source schemas describing *ingredients* used in African Traditional Medicine (ATM).

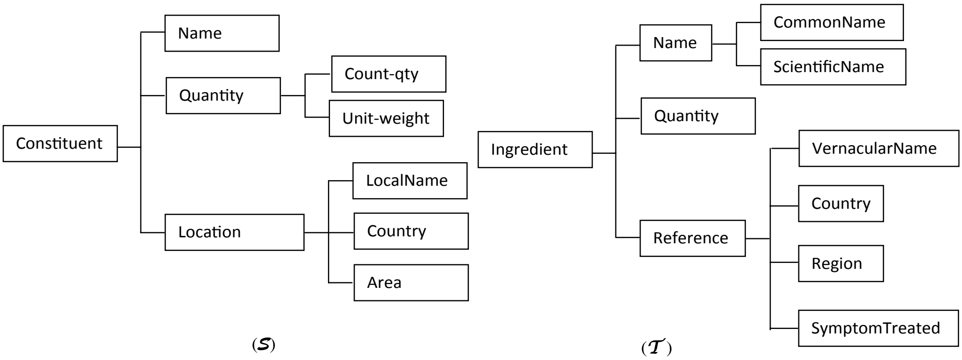

Fig. 2. Running example of 2 schemas in ATM.

Table 1. Example of possible mappings with their corresponding degree of reliability.

No	Possible mappings	$d(m_i)$
m_1	(Ingredient,Constituent),(Quantity, Quantity), (Name, Name), (Location, Reference)	0.252
m_2	(Count-qty, Quantity),(Name, CommonName), (Location, Country)	0.235
m_3	(Unit-weight, Quantity), (Name, ScientificName), (Location, Region)	0.230
m_4	(LocalName, VernacularName), (Country,Country)	0.176
m_5	(Area, Region)	0.106

The ingredients are usually the plant used to prepare *potion* (traditional-based drugs). It also shows common information usually collected about ingredient: a name which is either a scientific name, a common name or a vernacular name; the quantity and the unit used to quantify the ingredient is also specified; and the region where the ingredient can be found. Table 1 presents a set of 5 possible mutually distinct mappings, $m_i, i = 1, \dots 5$, with their corresponding reliable degrees $d(m_i)$. We are going to use these mappings as running example throughout this paper.

3.4. *Definitions and theorems*

We introduce below the formal definition of a reliable mapping, where $\mathcal{S}$ is a source schema and $\mathcal{T}$ a target schema.

Definition 1. A reliable mediated schema is a couple $(\mathcal{T}, d_{\mathcal{T}/\mathcal{S}})$, where $\mathcal{T}$ is a possible mediated schema and $d_{\mathcal{T}/\mathcal{S}}$ is the reliability degree of $\mathcal{T}$ with respect to $\mathcal{S}$.

Definition 2. A tag matching $\mathfrak{tag}(x_i, x_j)$ is a quadruple

$$(x_i, x_j, sim(x_i, x_j), op(x_i, x_j)),$$

where $sim(x_i, x_j)$ is the semantic similarity between x_i and x_j; and $op(x_i, x_j)$ is the structural similarity between x_i and x_j.

Example 1. From the running example Sec. 3.3, we may have the following tag matching between Ingredient and Constituent:
$\mathfrak{tag}(Ingredient, Constituent) = (Ingredient, Constituent, 0.78, \equiv)$.

Definition 3. A reliable mapping m is a couple $(m, d(m))$ where: m is a set of mutually distinct one-to-one tag matching results between elements of $\mathcal{S}$ and $\mathcal{T}$; $d(m)$ is the reliability degree of the mapping m with respect to $\mathcal{S}$.

Example 2. From the running example Sec. 3.3, we have the following reliable mapping:
$(m_1, d(m_1)) = ((Ingredient, Constituent, 0.78, \equiv), (Quantity, Quantity, 0.93, \equiv), (Name, Name, 0.93, \equiv), (Location, Reference, 0.51, \equiv), 0.25)$

Definition 4. A reliable mapping set $\mathcal{M}$ between $\mathcal{S}$ and $\mathcal{T}$ is a set of couples $\{(m_i, d(m_i)), i = 1 \cdots n\}$, where

(a) $\forall\, i = 1 \cdots n, (m_i, d(m_i))$ is a possible reliable mapping between $\mathcal{S}$ and $\mathcal{T}$

(b) $\forall\, i \neq j, m_i \neq m_j$
(c) $\sum_{i=1}^{n} d(m_i) = 1$

Example 3. From the running example Sec. 3.3, the set $M = \{(m_1, d(m_1)), (m_2, d(m_2)), (m_3, d(m_3)), (m_4, d(m_4)), (m_5, d(m_5))\}$ is reliable mapping set.

A reliable mapping set so defined is the set of mutually distinct reliable mappings between a source schema and a target schema. Assigning reliability degrees to the semantic mappings can help to overcome the uncertainty among the multiple mappings automatically produced. In order to manage the set of reliable mappings, a reliability degree threshold is generally used [4, 3]. Doing so lead to another level of uncertainty. In fact, as we discuss in [4], if the semantic mappings are mutually distinct, the value of reliability degrees then depends on the number of semantic mappings automatically produced by the system. Therefore, if the number of semantic mappings is too high, the value of reliability degrees will be too low; choosing a threshold might become an ambiguous or uncertain task.

For example, if we consider the mappings from the running example Sec. 3.3, if we choose a threshold $\theta = 0.15$, information hidden behind the mapping m_5 might be lost because $d(m_5) < \theta$.

To address this issue, we propose to lead the system treats all the available information using possibility theory [6]. Our proposed method, pos-mapping divides the set of reliable mappings into subsets of possibility distributions [9]. To construct these subsets, we first rank the elements of the set $\mathcal{M} = \{(m_i, d(m_i)), i = 1 \cdots n\}$ in descending order and add to that set two reliable mappings, $(m_{high}, d(m_{high}))$ and $(m_{low}, d(m_{low}))$ such that $d(m_{high}) = 1$ and $d(m_{low}) = 0$. We so obtain a well-ordered set $\mathcal{U}$ defined as follow.

Definition 5. A well-ordered set $\mathcal{U} = \{(u_1, d(u_1)), \ldots, (u_n, d(u_n))\}$ is a reliable mapping set $\mathcal{M} = \{(m_i, d(m_i)), i = 1 \cdots n\}$ such that

(a) $d(u_1) > d(u_2) > \cdots > d(u_n)$
(b) $(u_1, d(u_1)) = (m_{high}, d(m_{high}))$
(c) $(u_n, d(u_n)) = (m_{low}, d(m_{low}))$

Example 4. From the running example Sec. 3.3, we construct a well-ordered mapping set $\mathcal{U} = \{(u_1, d(u_1)), (u_2, d(u_2), \ldots, (u_7, d(u_7))\}$ as follows:

(1) $(u_1, d(u_1)) = (m_{high}, 1)$
(2) $(u_2, d(u_2)) = (m_1, 0.252)$
(3) $(u_3, d(u_3)) = (m_2, 0.235)$
(4) $(u_4, d(u_4)) = (m_3, 0.230)$
(5) $(u_5, d(u_5)) = (m_4, 0.176)$
(6) $(u_6, d(u_6)) = (m_5, 0.106)$
(7) $(u_7, d(u_7)) = (m_{low}, 0)$

Theorem 1. *Let* $\mathcal{U} = \{(u_1, d(u_1)), \ldots, (u_n, d(u_n))\}$ *be a well-ordered mapping set for a natural number* $n > 1$*, then* $1 - d(u_i) = d(u_{n+1-i})$*, for* $1 \leq i \leq n$*.*

Proof. The proof of Theorem 1 is done by induction on the index i of the elements of $\mathcal{U}$. Let $\mathcal{P}(i)$ be the property: $1 - d(u_i) = d(u_{n+1-i})$.

- For $i = 1$, we have:

$$\begin{aligned} 1 - d(u_1) &= 1 - 1 \\ &= d(u_n) \\ &= d(u_{n+1-1}) \end{aligned}$$

So $\mathcal{P}(1)$ holds.

- We now suppose that $\mathcal{P}(i)$ holds and prove that $\mathcal{P}(i+1)$ also holds:

$$\begin{aligned} d(u_{i+1}) &= d(u_{n-n+i+1}) \\ &= d(u_{n+1-(n-i)}) \\ &= 1 - d(u_{n-i}) \\ &= 1 - d(u_{n-i-1+1}) \\ &= 1 - d(u_{(n+1)-(i+1)}) \end{aligned}$$

Therefore, $\mathcal{P}(i+1)$ also holds. □

We now define a possibility degree function π from $\mathcal{U}$ to $[0, 1]$ such that there exists at least one element $u \in \mathcal{U}$ which is a correct mapping between the source schema and the target schema, i.e. $\pi(u) = 1$. To let the system manage all the available information, we divide the well-ordered set into subsets of possibility distributions. Let $\mathcal{S}$ be a source schema, $\mathcal{T}$ be a target schema, $\mathcal{U}$ the set of well-ordered mappings between $\mathcal{S}$ and $\mathcal{T}$, and A a subset of $\mathcal{U}$. We assign to A a characteristic function $\pi_A(u) \in [0, 1]$ which is the *possibility degree* of the element u being a correct mapping between $\mathcal{S}$ and $\mathcal{T}$; and we assumed that $\exists\, u \in A | \pi_A(u) = 1$, and, $\forall\, u \notin A$, $\pi_A(u) = 1 - d(u)$.

Definition 6. Let $\mathcal{U} = \{(u_1, d(u_1)), \ldots, (u_n, d(u_n))\}$ be a well-ordered set such that $d(u_1) > d(u_2) > \cdots > d(u_n)$. We define n subsets $A_i, 1 \leq i \leq n$ of $\mathcal{U}$ and their corresponding *characteristic function* π_{A_i} as follows:

$$\begin{cases} A_i = \{u_1, u_2, \ldots, u_i\} \\ \pi_{A_i}(u_j) = \begin{cases} 1, & \text{if } u_j \in A_i \\ 1 - \displaystyle\sum_{k=i+1}^{j} d(u_k), & \text{otherwise} \end{cases} \end{cases} \tag{4}$$

Example 5. Considering the well-ordered mapping set built in Example 4, we may construct the following subset of possibility distributions A_1 and A_2:

$$\begin{cases} A_1 = \{u_1\} \\ \pi_{A_1}(u_1) = 1 \\ \pi_{A_1}(u_2) = 1 - d(u_2) = 0.748 \\ \pi_{A_1}(u_3) = 1 - d(u_2) - d(u_3) = 0.513 \\ \pi_{A_1}(u_4) = 1 - d(u_2) - d(u_3) - d(u_4) = 0.283 \\ \pi_{A_1}(u_5) = 1 - d(u_2) - d(u_3) - d(u_4) - d(u_5) = 0.107 \\ \pi_{A_1}(u_6) = 1 - d(u_2) - d(u_3) - d(u_4) - d(u_5) - d(u_6) = 0.001 \\ \pi_{A_1}(u_7) = 1 - d(u_2) - d(u_3) - d(u_4) - d(u_5) - d(u_6) - d(u_7) = 0.001 \end{cases} \tag{5}$$

and

$$\begin{cases} A_2 = \{u_1, u_2\} \\ \pi_{A_2}(u_1) = 1 \\ \pi_{A_2}(u_2) = 1 \\ \pi_{A_2}(u_3) = 1 - d(u_3) = 0.765 \\ \pi_{A_2}(u_4) = 1 - d(u_3) - d(u_4) = 0.535 \\ \pi_{A_2}(u_5) = 1 - d(u_3) - d(u_4) - d(u_5) = 0.359 \\ \pi_{A_2}(u_6) = 1 - d(u_3) - d(u_4) - d(u_5) - d(u_6) = 0.253 \\ \pi_{A_2}(u_7) = 1 - d(u_3) - d(u_4) - d(u_5) - d(u_6) - d(u_7) = 0.253 \end{cases} \tag{6}$$

Following this examples, we may also construct the subsets of possibility distribution A_3, A_4, A_5, A_6, and A_7 with their corresponding characteristic functions π_{A_3}, π_{A_4}, π_{A_5}, π_{A_6}, and π_{A_7} for our running example presented in Sec. 3.3.

We now study how the new characteristic function π_{A_i} behaves compared to the reliability degree function $d(u)$, for some $A_i \subseteq \mathcal{U}$.

Theorem 2. *Let A_i be the i^{th} subset of $\mathcal{U}$ as defined in Eq. 4; and $u_{j_1}, u_{j_2} \in \mathcal{U}$ such that at least one of them does not belong to A_i. Then $\pi_{A_i}(u_{j_1}) > \pi_{A_i}(u_{j_2})$ if and only if $d(u_{j_1}) > d(u_{j_2})$.*

Proof. Suppose that $d(u_{j_1}) > d(u_{j_2})$. we must prove that $\pi_{A_i}(u_{j_1}) > \pi_{A_i}(u_{j_2})$. We distinguish two cases:

- $u_{j_1} \in A_i$ and $u_{j_2} \notin A_i$: By Eq. 4, we have $j_1 < j_2$ and $\pi_{A_i}(u_{j_1}) = 1$ and $\pi_{A_i}(u_{j_2}) = 1 - \sum_{k=i+1}^{j_2} d(u_k) < 1$ (because $0 < \sum_{k=i+1}^{j_2} d(u_k) < 1$).
- $u_{j_1}, u_{j_2} \notin A_i$: By Eq. 4, we have $\pi_{A_i}(u_{j_1}) = 1 - \sum_{k=i+1}^{j_1} d(u_k)$ and $\pi_{A_i}(u_{j_2}) = 1 - \sum_{k=i+1}^{j_2} d(u_k)$. From Definition 6, $d(u_{j_1}) > d(u_{j_2})$ implies $j_1 < j_2$, which implies $\sum_{k=i+1}^{j_1} d(u_k) < \sum_{k=i+1}^{j_2} d(u_k)$, which implies $1 - \sum_{k=i+1}^{j_1} d(u_k) > 1 - \sum_{k=i+1}^{j_2} d(u_k)$, i.e. $\pi_{A_i}(u_{j_1}) > \pi_{A_i}(u_{j_2})$.

Reciprocally, suppose that $\pi_{A_i}(u_{j_1}) > \pi_{A_i}(u_{j_2})$. we must prove that $d(u_{j_1}) > d(u_{j_2})$. There are also two cases:

- $u_{j_1} \in A_i$ and $u_{j_2} \notin A_i$: By Eq. 4, this implies that $j_1 < j_2$. By Definition 6, we have $d(u_{j_1}) > d(u_{j_2})$.
- $u_{j_1}, u_{j_2} \notin A_i$: By Eq. 4, we have $\pi_{A_i}(u_{j_1}) = 1 - \sum_{k=i+1}^{j_1} d(u_k)$ and $\pi_{A_i}(u_{j_2}) = 1 - \sum_{k=i+1}^{j_2} d(u_k)$. We have $\pi_{A_i}(u_{j_1}) > \pi_{A_i}(u_{j_2})$ implies $1 - \sum_{k=i+1}^{j_1} d(u_k) > 1 - \sum_{k=i+1}^{j_2} d(u_k)$, which implies $\sum_{k=i+1}^{j_1} d(u_k) < \sum_{k=i+1}^{j_2} d(u_k)$, which implies $j_1 < j_2$, and therefore $d(u_{j_1}) > d(u_{j_2})$. □

Literally, Theorem 2 above asserts that the reliability degree function d and the possibility degree function π yield the same ordering of the elements of $\mathcal{U}$. That is, the higher the reliability degree of an element, the higher its possibility degree and vice versa. Therefore, the reliability degree and the information carried by an element are conserved in its possibility degree. Moreover, the possibility degree of a given element u_i can be computed using the possibility degree of the element u_{i-1} as showed in the following Lemma.

Lemma 1. *Given a well-ordered set $\mathcal{U}$ of mappings between two schemas $\mathcal{S}$ and $\mathcal{T}$, $\mathcal{U} = \{(u_1, d(u_1)), \dots, (u_n, d(u_n))\}$ such that $d(u_1) > d(u_2) > \cdots > d(u_n) \geq 0$; $\mathcal{U}$ can be divided into subsets $A_{i,1\leq i\leq n}$, such that for $1 \leq j \leq n$,*

$$\begin{cases} A_1 = \{u_1\} \\ \pi_{A_1}(u_1) = 1 \\ \pi_{A_1}(u_j) = 1 - \sum_{k=2}^{j} d(u_k) \\ A_i = A_{i-1} \cup \{\ u_i\} \\ \pi_{A_i}(u_j) = \begin{cases} 1, & \text{if } u_j \in A_i \\ \pi_{A_{i-1}}(u_j) + d(u_i), & \text{otherwise} \end{cases} \end{cases} \tag{7}$$

Proof. From Eq. 4, there are three cases:

- $i = 1$: the result is straightforward.
- $u_j \in A_i$: then $\pi_{A_i}(u_j) = 1$.
- $u_j \notin A_i$: then

$$\begin{aligned} \pi_{A_i}(u_j) &= 1 - \sum\nolimits_{k=i+1}^{j} d(u_k) \\ &= 1 - \sum\nolimits_{k=i+1}^{j} d(u_k) - d(u_i) + d(u_i) \\ &= 1 - \sum\nolimits_{k=i}^{j} d(u_k) + d(u_i) \\ &= 1 - \sum\nolimits_{k=(i-1)+1}^{j} d(u_k) + d(u_i) \\ &= \pi_{A_{i-1}}(u_j) + d(u_i) \end{aligned}$$

□

Example 6. Considering the subsets A_1 and A_2 constructed in Example 5, we have:

$$\begin{cases} A_2 = \{u_1, u_2\} = A_1 \cup \{u_2\} \\ \pi_{A_2}(u_1) = 1 \\ \pi_{A_2}(u_2) = 1 \\ \pi_{A_2}(u_3) = \pi_{A_1}(u_3) + d(u_2) = 0.765 \\ \pi_{A_2}(u_4) = \pi_{A_1}(u_4) + d(u_2) = 0.535 \\ \pi_{A_2}(u_5) = \pi_{A_1}(u_5) + d(u_2) = 0.359 \\ \pi_{A_2}(u_6) = \pi_{A_1}(u_6) + d(u_2) = 0.253 \\ \pi_{A_2}(u_7) = \pi_{A_1}(u_7) + d(u_2) = 0.253 \end{cases} \tag{8}$$

We may introduce the following result.

Theorem 3. *The function $\pi_{A_i}(u_j)$ denoting the possibility degree of the element u_j being a correct mapping between the source schema $\mathcal{S}$ and the target schema $\mathcal{T}$ in the reliable mapping set A_i is a recursive function.*

Proof. The proof of the Theorem 3 follows from Lemma 1 demonstrated above. □

Literally, the consequence of the theorem 3 above is that the information available is managed step by step starting from the most pertinent information to the less pertinent one. At each step of the process, the current information is enriched with the information managed at the previous step. Considering the fact that some mappings automatically provided might be partially correct, using a recursive function will piece together two or many partially correct mappings into one better mapping. Moreover, from time to time as the system is being used, the system would incrementally construct a possible mapping which is closer to "the correct mapping" and therefore provide "better results" to users posed queries or keyword searches. We now introduce in the following subsection a discussion between the existing methods and the method pos-mapping proposed in this paper.

4. Experiments

4.1. *Experimental setup*

Based on the method presented above, we built a new system named KSpace++. We used an XML enabled Oracle database to store our data and we implemented our methods, algorithms in C++. We conducted our experiments on a mixed network with three computers running on Window, Linux-Fedora 10 and Ubuntu Desktop 9, respectively. Each computer using 2 CPUs Intel Pentium M 3Ghz with 2Gb memory. We evaluated KSpace++ by computing the Precision, Recall and F-measure of the results provided when using pos-mapping method. Indeed, considering a given result-set $\mathcal{R}$ obtained using a given method and $\mathcal{R}_{ex}$ a non-empty Result-set provided by an expert or a user, these metrics are calculated as follows:

Precision: expresses the proportion of expected results among the results produced using a given method.

$$Precision = \frac{|\mathcal{R} \cap \mathcal{R}_{ex}|}{|\mathcal{R}|}$$

Recall: shows the proportion of correct results extracted by the system, as a fraction of the expected results.

$$Recall = \frac{|\mathcal{R} \cap \mathcal{R}_{ex}|}{|\mathcal{R}_{ex}|}$$

F-measure: is a compromise between recall and precision.

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Pecall}$$

4.2. *Experimental results*

We first evaluate the feasibility of our system using real world data sources in African Traditional Medicine (ATM) domain using 3 scenarios: plants, diseases and treatments as shown in Fig. 3. The sources were selected from different projects [10–13] on ATM from diverse African countries. Each sub-domain (plant, disease or treatment) contains hundreds of documents.

We further observe how the precision behaves when Kspace and KSpace++ manage less to many reliable mappings. We then present in Fig. 4 how the metrics behaves when the number of reliable schemas managed increases.

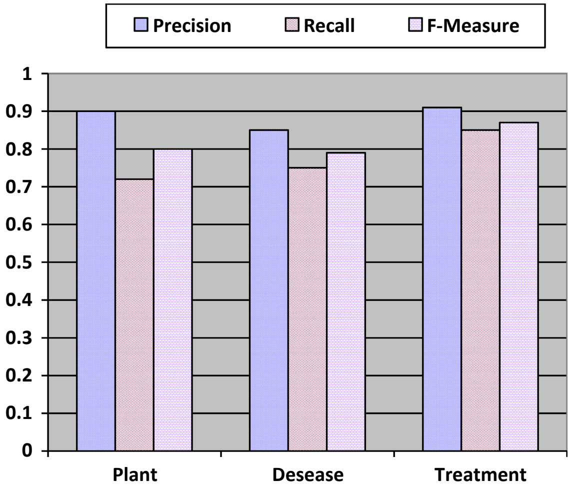

Fig. 3. KSpace++ Precision, Recall and F-measure on the results provided.

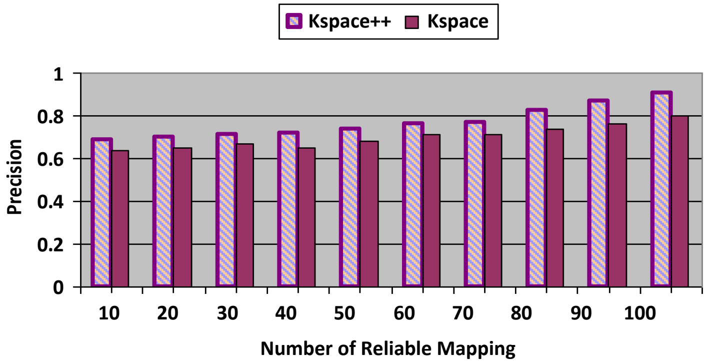

Fig. 4. Kspace and KSpace++ Precision when the number of reliable mappings increases.

From Fig. 4, we can note that the Kspace++ results seem to become more and more accurate when the number of reliable schema managed increases while KSpace's results varies constantly.

We continue with the efficiency evaluation of KSpace++ by observing its response time of the number of input sources as showed in Fig. 5.

We may observe that the response time is a linear function on the number of input sources.

We finally evaluate the performance of KSpace++. In fact, we compare the response time obtained in the new system KSpace++ with the response time provided in KSpace system and with the existing system UDI built in [3] based on their method called pmapping. The results are shown in Fig. 6.

From Fig. 6, we may observe that, KSpace++ is slightly more efficient than KSpace and UDI. In fact, for less than a 150 input schemas, KSpace and KSpace++ appears faster than UDI, and for more than 150 input schema, KSpace is slightly

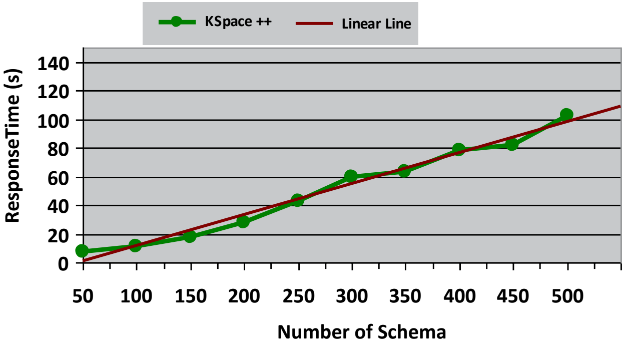

Fig. 5. KSpace++ response time on the number of input schema (Plant scenario): *The response time is a linear function on the number of input sources.*

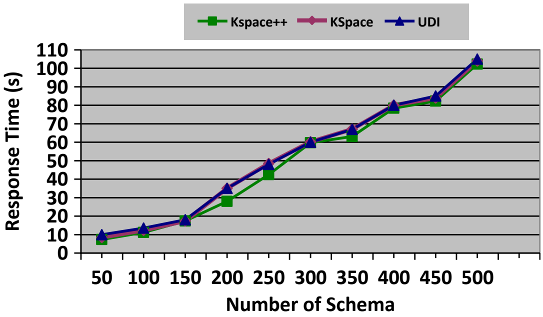

Fig. 6. KSpace++ (pos-mapping) compared with KSpace(rMedMap) and UDI (pmapping).

faster than UDI while KSpace++ keeps its time inscreasing linearly yet still faster than the others.

5. Related Work and Discussion

We present in this section some existing methods which also manage uncertainty between the mappings automatically provided in a DataSpace Support Platform (DSSP). We further argue about the fact that a system might have the possibility to manage "all" the mappings provided or to bring together two mappings partially correct.

5.1. *Related work*

There exists a number of methods and systems built to manage uncertainty in the data integration field. The existing methods are usually proposed relatively to a specific type of schema such as structural data base, XML enabled systems and semantic Web systems. For example AQUA [14] answers queries over heterogeneous sources described by their ontologies and DSSim [15] is an algorithm to deal with the incomplete and inconsistency in the mapping generated between two ontologies, with the aim to integrate information on the Semantic Web. The reader can refer to [16] for a survey on ontology mapping systems. Authors also proposed in [17] an extension of current practice in schema matching with the simultaneous use of top-K schema mappings rather than a single best mapping. Similarly, in [18, 19] authors introduce a novel data structure called block tree to manage possible mappings between two heterogeneous XML schemas. To manage uncertainty, they evaluate the Probabilistic Twig Query (PTQ), which returns the probability of a portion of an XML document that matches the query pattern and returns the top-k PTQ, the k-highest mappings. The reader may refer to [3, 4, 20–27] for more recent methods on schema mappings.

We now address the problem of uncertainty for data integration in DataSpace Supports Platforms (DSSP). As we stated in the previous sections, in a dataspace space support platforms, the system self-sustained and self-managed its mappings.

The system therefore needs to support uncertainty among its mediated schemas and its semantic mappings. Unfortunately, between the mediated schemas or semantic mapping automatically provided; some of the mappings might be correct and others partially correct. Dong *et al.* [26] proposed the concept of probabilistic schema mapping for data integration with uncertainty in DSSP. Sarma *et al.* further introduced p-mapping [3], a probabilistic-based method to manage uncertainty between the semantic mappings automatically generated between a source schema and a target mediated schema. In fact, for a given mapping, pmapping computes the probability measure of the given mapping compared to the possible mapping. pmapping further chooses a threshold among which the probable mapping can be managed by the system.

Kuicheu *et al.* [4] introduced a new method which assigned reliability degree to the mappings automatically provided with the aim to manage uncertainty among the mappings. In fact, rMedMap computes reliability measure of a given possible mapping compared to a given source schema and compared to the whole set of available source schemas. The reliable source further managed are the ones which their reliability degrees are greater than a certain chosen threshold. Therefore, in both pmapping and rMedMap, the system needs to choose a certain threshold at a given time. With the number of reliable or probable mappings provided, choosing a threshold might become an uncertain task and may lead to information lost. Then, Assigning reliability or probability to the possible mappings is a necessary step; but it is not sufficient to manage uncertainty among the mappings.

5.2. *Discussion*

In both pmapping and rMedMap presented above, the system needs to choose a certain threshold at a given time. With a high amount of reliable mappings provided, choosing a threshold might become an uncertain task and may lead to information lost. Then, it appears that a system which can manage all the available mappings might return its best endeavor result to user's posed query or keyword search. Assigning reliability or probability to the possible mappings then appears as a necessary step though but not sufficient enough to manage uncertainty among all the available mappings.

We propose in this paper, the pos-mapping method which manages all the available mappings step by step from the most reliable mapping to the less reliable one. The pos-mapping method computes the reliability degrees of the available mappings using the function proposed in [4]. Then instead of choosing a threshold, the pos-mapping first made the set of reliable mappings into a descending order. It further divides the well-ordered reliable mappings into subsets of possibility distributions and assigns to each subset a possibility degree computed using the reliability degree of the mappings belonging to the given subset. We show that the possibility degree function is correlated to the reliability function. We finally show that the possibility function is a recursive function. The recursiveness of the possibility degree function enables the system to manage automatically all the available reliable mappings from the most reliable mapping to the less reliable one. That is, when the system is managing the current mapping, it uses the result obtained when managing the latter mapping. Using such a method has two-fold advantages: First, step by step the system has the possibility to manage all the available mappings; Second, the system may incrementally bring two or more partially correct mappings into one correct mapping. We think that using pos-mapping enables a system to provide its best endeavor results to a user's posed query or keyword search.

6. Conclusion and Future Works

We present in this paper *pos-mapping*, a possibility theory based method for managing reliable mappings automatically provided from a set of independently constructed source schemas in DataSpace Support Platforms. Our main objective was to let the system self-manage all the available information contained in the automatically provided mappings. The purpose of pos-mapping method is to enable a DSSP to build "a correct mapping" when possible or to combine two or many mappings partially correct into a "better mapping" in order to provide its best endeavor results to a user posed query. We therefore propose a method which first ordered and then divided the set of reliable mappings into subsets of possibility distributions. We finally show how the system can manage the subsets of possibility distributions recursively. In other words, the reliable mappings are managed incrementally from the most reliable mapping to the less reliable one and the results of the previous reliable mappings managed is used in the reliable mapping currently managed. Experimental results show that the results provided by our system is more and more accurate as the number of reliable schemas managed increases and compared to existing systems, our system seems more efficient and may lead the system to managing all the available information existing in the reliable mappings.

Acknowledgments

This work is supported by National Natural Science Foundation of China (Grant No. 61370060), Jiangsu Provincial Natural Science Foundation of China (No. BK2011454) and the Open Foundation from Computer Application Technology in the Most Important Subjects of Zhejiang (No. 10Y0001).

We are grateful to the anonymous reviewers whose comments and suggestions helped us to improve the quality of this paper.

References

1. M. Franklin, A. Halevy and D. Maier, From databases to dataspaces: A new abstraction for information management, *ACM SIGMOD Record* **34**(4) (2005) 27–33.

2. A. Halevy, M. Franklin and D. Maier, Principles of dataspace systems, in *Proc. 25th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, Chicago, USA, 2006, pp. 1–9.
3. A. Das Sarma, L. Dong and A. Halevy, Bootstrapping pay-as-you-go data integration systems, in *Proc. of the 28th ACM SIGMOD*; Vancouver, Canada, 2008.
4. N. C. Kuicheu, N. Wang, G. N. Fanzou Tchuissang, D. Xu, G. Dai and F. Siewe, Managing uncertain mediated schema and semantic mappings automatically in dataspace support platforms, *Computing and Informatics* **32** (2013) 175–202.
5. D. Dubois, Possibility theory and statistical reasoning, in *Computational Statistics and Data Analysis* **51** (2006) 47–69.
6. D. Dubois and H. Prade, When upper probabilities are possibility measures, *Fuzzy Sets and Systems* **49** (1992) 65–74.
7. S. M. Ross, *A First Course in Probability* (Prentice Hall, 1998).
8. W. Spohn, Ordinal conditional functions: A dynamic theory of epistemic states, *Causation in Decision, Belief Change and Statistics*, 1988, pp. 105–134.
9. L. A. Zadeh, Fuzzy sets as a basis for a theory of possibility, *Fuzzy Sets and Systems* **1** (1978) 3–28.
10. L. P. Fotso, Table of entities and attributes of data bases in MEDITRA, in *Rapport de recherche*, Number 20, University of Yaounde 1, February 1999 (in French).
11. MetAfro: http://www.metafro.be.
12. AFRITRADOMEDIC: http://www.afritradomedic.com.
13. MRC: http://www.mrc.ac.za.
14. M. Nagy, M. Vargas-Vera and E. Motta, Multi-agent ontology mapping framework in the AQUA question answering system, in *Fourth International Mexican Conference on Artificial Intelligence*, LNAI Vol. 3789, 2005, pp. 70–79.
15. M. Nagy, M. Vargas-Vera and E. Motta, DSSim-ontology mapping with uncertainty, in *Proc. of 1st International Workshop on Ontology Matching*, Athens, Georgia, USA.
16. N. Choi, I.-Y. Song and H. Han, A survey on ontology mapping, *SIGMOD Record* **35**(3) (2006) 34–41.
17. G. Avigdor, Managing Uncertainty in schema matching with top-K schema mappings, *Journal on Data Semantics VI*, LNCS Vol. 4090, 2006, pp. 90–114.
18. J. Gong, R. Cheng and D. W. Cheung, Efficient management of uncertainty in XML schema matching, in *VLDB Journal: The International Journal on Very Large Data Bases Archive* **21**(3) (2012) 385–409.
19. J. Gong, R. Cheng and D. W. Cheung, Managing uncertainty of XML schema matching, in *International Conference on Data Engineering*, 2010, pp. 297–308.
20. H. H. Do and E. Rahm, Matching large schemas: Approaches and evaluation, *Journal of Information Systems* **32**(6) (2007) 857–885.
21. M. Arenas and L. Libkin, XML data exchange: Consistency and query answering, *Journal of the ACM* **55**(2) (2008).
22. K. Saleem and Z. Bellahsene, PORSCHE: Performance ORiented SCHEma Matching, in *Proc. of 16th ACM International World Wide Web Conference*, Banff, Canada, May 27–28, 2007.
23. M. Magnani and D. Montesi, Uncertainty in data integration: Current approaches and open problems, in *Proc. of International Conference on Very Large DataBase*, 2007.
24. B. He and K. C. Chang, Statistical schema matching across web query interfaces, in *Proc. of ACM SIGMOD*, 2003.

25. A. Doan, J. Madhavan, P. Domingos and A. Y. Halevy, Learning to map between ontologies on the semantic web, in *Proc. of the International World Wide Web Conference*, 2002.
26. X. Dong, A. Y. Halevy and C. Yu, Data integration with uncertainty, in *Proc. of International Conference on Very Large DataBase*, 2007.
27. X. Dong, Providing Best-Efforts Services in Dataspace Systems, PhD Dissertation, University of Washington, 2007.