Computing and Informatics, Vol. 32, 2013, 175–202

MANAGING UNCERTAIN MEDIATED SCHEMA AND SEMANTIC MAPPINGS AUTOMATICALLY IN DATASPACE SUPPORT PLATFORMS

Nathalie Cindy Kuicheu, Ning Wang*
Gile Narcisse Fanzou Tchuissang, De Xu

School of Computer and Information Technology, Beijing Jiaotong University
3 Shangyuancun Xizhimenwai, 100044 Beijing China
e-mail: nathkuicheu@yahoo.fr, nwang@bjtu.edu.cn
fanzounar2002@yahoo.fr, dxu@bjtu.edu.cn

Guojun Dai

Computer School, Hangzhou Dianzi University
Hangzhou, 310018, China
e-mail: daigj@hdu.edu.cn

Francois Siewe

Software Technology Research Laboratory, Faculty of Technology
De Montfort University, The Gateway, Leicester LE1 9BH, United Kingdom
e-mail: fsiewe@dmu.ac.uk

Communicated by Ulrich Eisenecker

Abstract. Contrary to existing heterogeneous data integration systems which need to be fully integrated before using, a Dataspace Support Platform is a self-sustained system which automatically provides for the user its best endeavor results regardless of how integrated its sources are. Therefore, a Dataspace Support Platform needs to support uncertainty in mediated schema and in schema mappings. This paper proposes a novel approach to automatically providing reliable mediated schemas and reliable semantic mappings in Dataspace Support Platforms. Our aim is to increase the system's endeavor results by leading it to considering as much as pos-

* correspondence author

sible information available in any source connected. In fact, we first extract from the source schemas, their corresponding graph representations. Then, we introduce algorithms which automatically extract a set of mediated schemas from the graph representations and a set of semantic mappings between a source and a target mediated schema. Finally, we assign reliability degrees to the mediated schema generated and to the semantic mappings. Indeed, the higher the reliability degree of a given mediated schema or semantic mapping, the more consistent with the source it is. Compared with existing systems, experimental results show that our system is faster and, although completely automatic, it produces reliable mediated schemas and reliable semantic mappings which are as accurate as those produced by semi-automatic systems.

Keywords: Schema matching, mediated schema, semantic mappings, reliability degrees, dataspace

Keywords: 97R50, 68P15, 68P20, 03E75, 03E35

1 INTRODUCTION

Dataspace Management Systems [11] describe a platform supporting dataspaces where a dataspace contains a set of participants from heterogeneous sources of data; Relational Data Base, XML repository, Excel spreadsheets; and a set of relationships between those heterogeneous participants. Existing heterogeneous data integration systems require a full integration of its sources before any service can be provided. Hence the data integration system knows the precise relationships between the terms used in each schema. As a result, significant up-front effort is required in order to set up a data integration system.

Contrary to traditional data integration systems, a DataSpace Support Platform (DSSP) [10] is a self-sustained system which self-produced its mediated schemas and schema mappings, self-managed uncertainty among them and which can be improved incrementally as the system is used or as new sources get connected to the platform. Hence, the results provided to a posed query are its best endeavor results. We therefore need to perform an automatic schema matching, extract some useful relationships between the sources, provide possible mediated schema and semantic mappings and manage uncertainty among them.

Several methods have been proposed to producing semi-automatically single mediated schema for data integration in XML enabled or relational database management systems. For example COMA [3] is a system which provides semantic mappings between two input schemas by combining multiple matchers [17]. Another system, Similarity Flooding [16] takes two graphs schemas as input, and produces as output a mapping between corresponding nodes of the schemas. PORSCHE [20] provides single mediated schema from a set of multiple tree-based input schemas. Some of

these systems take as input only two source schemas while others take multiple schemas but they all generate as output a single mediated schema generally with feedbacks from domain experts. Unfortunately, in the possible application domain of dataspace, the user might not be skilled enough to manipulate mappers or to provide accurate mappings. Moreover, the source schemas connected to a dataspace are of heterogeneous structure, that is, they could be files, relational databases, XML repositories, web pages and so on. That is why new solutions are needed for data integration in DSSP.

Authors recently introduced probabilistic schema mappings [2] for data integration with uncertainty in DSSP. Their method describes a probability over a set of possible mediated schemas between relational tables. The probability of a given mediated schema is computed as the ratio between the number of sources the mediated schema is consistent with and the total number of sources. Then, the mediated schema which has the highest probability is the one consistent with the highest number of sources. In other words, the less a mediated schema is consistent with a set of sources, the less the information contained in those sources will be used. To improve the best effort services [8] produced with probabilistic mapping, we propose to lead the system self-produce *best endeavor results* incrementally by assigning degree of consistency or reliability degree to the mediated schema with respect to each source and with respect to the whole set of sources.

In fact, since we have several types of schema in dataspace, we first propose to extract from the source schemas their corresponding graph representations; schemas are therefore viewed in our system, referred here as KSpace (Knowledge Space), as graph structures containing terms and their inter-relationships. After that, we present an algorithm which automatically extracts a set of mediated schemas from the graph representations; and, instead of finding whether a given mediated schema is consistent with a source, we compute its reliability degree with respect to the source. Indeed, the higher the reliability degree of a given mediated schema or semantic mapping, the more consistent with the source it is. In short our contributions are as follows:

1. To our knowledge, this is the first work that addresses the problem of assigning reliability degrees to a set of possible mediated schemas to managing uncertainty in a self-sustained based system.
2. We designed equations to compute reliability degrees of each mediated schema with respect to a source schema and with respect to the whole set of sources. These equations depend on the semantic similarity measure between terms used in both the mediated schema and the source schema.
3. We propose a matching and mapping method named rMedMap which provides reliable mediated schema (rMed) and reliable mappings (rMap) from a set of independently constructed source schemas. We argue about the type of relationship (mutually distinct or not) between the mediated schemas or mappings generated.

4. We implement the proposed method and evaluate its feasibility, efficiency, accuracy and performance using extensive experiments.

The rest of the paper is organized as follows: the system overview is introduced in Section 2, Section 3 describes rMedMap method, Section 4 presents the experiments, Section 5 discusses the related works and Section 6 concludes the paper.

2 SYSTEM OVERVIEW

2.1 Motivation

We introduce in this section the motivations which lead us to propose a reliability measure instead of using an existing measure such as a probability measure. Let us first recall that one of the most important aims to building a DSSP is to provide the user with basic functionalities like information retrieval or keyword search over all the sources connected without any domain experts intervention. In other words, the result of the system's endeavor should be the best which can be produced; then the system should exploit as much information as possible that is available in the sources. Assigning probabilities to mappings or mediated schema is equivalent to computing the ratio between the number of source a given mediated schema is consistent with and the total number of sources. Then to compute the numerator of this fraction, the system should determine whether a given mediated schema is consistent with a source. Hence, the mapping or mediated schema with the highest probability will be the one which is consistent with the highest number of sources. But, a given mediated schema could be consistent with a low number of sources, therefore its probability will be too low and a system constructed on such type of measure might not consider the information available in that source; this might lead to information loss. Therefore, we propose to measure the "*degree of consistency*" or "*reliability degree*" of a mediated schema with respect to a source; the system can then easily choose the convenient mediated schema when dealing with a given source. We think that using such measures could lead us to "*best endeavor results*".

For example, let us consider a scenario of 15 source schemas and 4 mutually distinct possible mediated schemas as shown in Figure 1. In this figure, a line from a mediated schema M_j to a source schema $\mathcal{S}_i$ means that the mediated schema M_j is consistent with the connected source schema $\mathcal{S}_i$. Therefore, the reliability degree of the mediated schema M_j with respect to the source $\mathcal{S}_i$ noted $d_{M_j/\mathcal{S}_i}$ is greater than a certain threshold θ. If we choose for example, $\theta = 0.2$ then the probability of the mediated schema M_4: $p(M_4) = \frac{1}{15} = 0.067 \leq \theta$ and the information in $\mathcal{S}_9$ might not be considered by a probability based system while it will be considered in a reliability based system because $d_{M_4/\mathcal{S}_9} \geq \theta$.

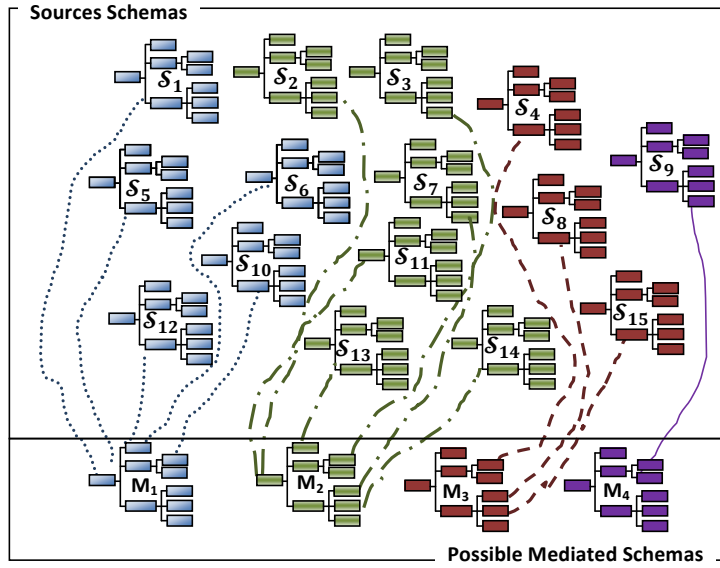

Fig. 1. Motivating example of 15 source schemas with 4 possible mediated schemas

2.2 System Architecture

Figure 2 presents the system architecture. From the sources connected to the DSSP, the system first extracts their corresponding internal graph representation. After that, the syntactic, semantic and structural matching can be carried out in order to deduce semantic and structural relationships between the graph representations. Then, the semantic and structural relationships are merged together and possible mediated schemas are produced. Finally, the system computes reliability degrees of the mediated schemas and reliable semantic mappings can be generated and transmitted to the query manager.

2.3 Running Example

The possible application domains of dataspace includes Personal Information Management, Web-Scale Information Management and Medical Information Management [11]. 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 3 shows an example of two source schemas describing *ingredients* used in African Traditional Medicine (ATM). The ingredients are usually the plant used to prepare *potion* (traditional-based drugs). Figure 3 shows common information usually collected about ingredient: a name which is either a scientific name, a common name or a vernacular name;

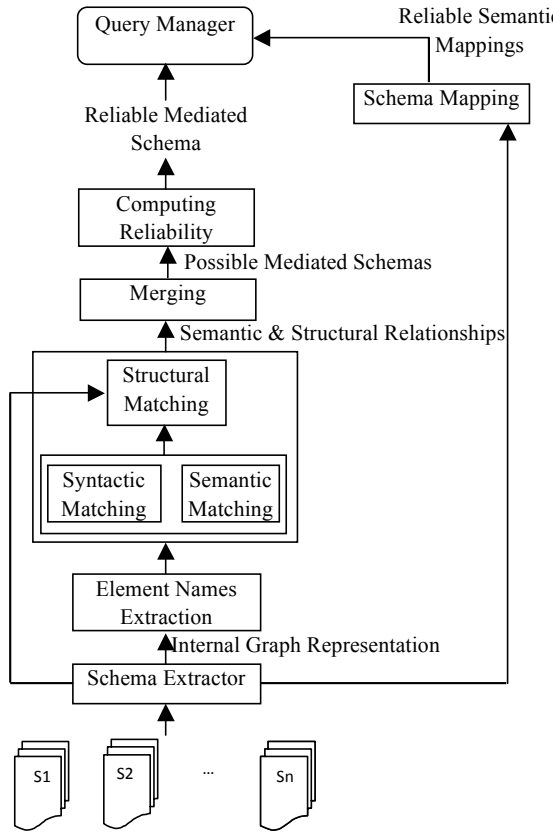

Fig. 2. System architecture

the quantity and the unit used to quantify the ingredient is also specified; and the region where the ingredient can be found. We are going to use these two schemas as running example throughout this paper.

3 RELIABLE SCHEMA MATCHING AND MAPPING PRINCIPLE (RMEDMAP)

In this section, we show how our system performs the matching between the sources. We also present algorithms used to produce reliable mediated schema between the source schemas. We formally define a source schema $\mathcal{S}$ as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ where $\mathcal{V}$ is the set of element names belonging to the source schema and $\mathcal{A} \subset \mathcal{V} \times \mathcal{V}$ is the set of parent-child relationships between element names. In the

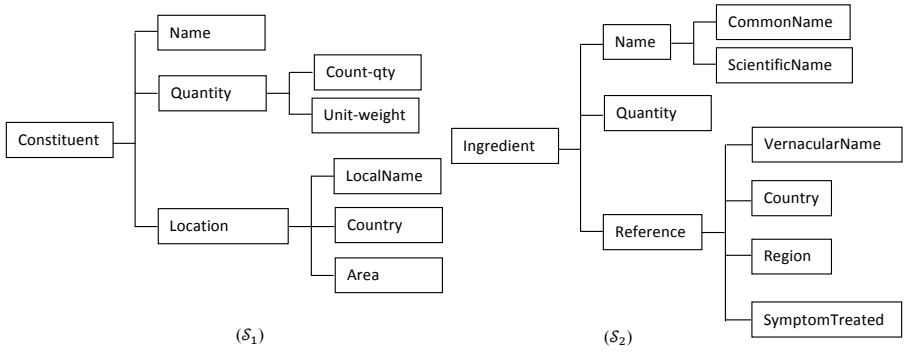

Fig. 3. Running example of 2 source schemas in ATM

following subsection we first present the notion of similarity measure between two element names.

3.1 Information Content Based Similarity Measure

There exist several formulas to compute semantic similarity between element names [21, 1, 14, 18, 22]. We are not going to improve one of them or to propose a new formula, but we are going to show how semantic similarity is computed based on the notion of Information Content (IC). When most of the proposed semantic similarities are reflexive, the information content based ones are symmetric and transitive [22]; this justifies our choice of an Information Content based formula. Therefore, one of the existing information content based formulas can be used in the sequel to compute the semantic similarity measure noted $sim(x_i, x_j)$ between two element names x_i and x_j.

Most of the existing semantic similarities using IC are improvements of the similarity measure proposed by Resnik in [18]; with the results of the similarity measure ranging from 0 for terms without similarity to 1. Resnik's proposed similarity measure uses a taxonomy or an ontology with multiple inheritance (subsumption relationships) as the representation model.

In other words, Resnik's idea for computing similarity measure using a taxonomy or an ontology is based on the fact that the more the probability of a concept increases the more concepts it subsumes, i.e. the higher it is within the taxonomy. The information content of a concept can be computed from its probability. By analogy to information theory [19], Resnik defines the information content of a concept as the negative logarithm of its probability; i.e. for a given concept c, $IC(c) = -\log(p(c))$. The Resnik's similarity of two concepts c_i and c_j noted $sim_{Resnik}(c_i, c_j)$ is the maximal information content of all concepts subsuming both c_i and c_j: $sim_{Resnik}(c_i, c_j) = \max_{c \in s(c_i,c_j)}[-\log(p(c))]$; where $s(c_i, c_j)$ is the set of concept subsuming both c_i and c_j.

We therefore introduce the property $\mathcal{P}$ defined as follows: if $c_i \equiv_{\delta_1} c_j$ and $c_j \equiv_{\delta_2} c_k$ then $c_i \equiv_{\min(\delta_1,\delta_2)} c_k$; where the notion of "$\delta$ equivalent" noted $\equiv_\delta$ and respecting the property $\mathcal{P}$ is defined as follows: the similarity measure between two given elements x_i and x_j noted $sim(x_i, x_j)$ is equal to δ means that the information content of their corresponding concepts c_i and c_j in the taxonomy are δ equivalent and we note $IC(c_i) \equiv_\delta IC(c_j)$.

We then use this similarity measure to perform one-to-one matching by computing the tag matching between element nodes names as presented in the following subsection.

3.2 Reliable Matching Process

Let $\mathcal{S}_i$ and $\mathcal{S}_j$ be two source schemas, let x_i, x_j be two non-leaf element nodes of $\mathcal{S}_i$ and $\mathcal{S}_j$, respectively. Non-leaf element nodes are concepts collected from the corresponding graph representiations of the source schema which are not leaf on the graph.

Definition 1. x_i and x_j are said to be *highly similar* and we note $x_i \simeq x_j$, if $sim(x_i, x_j) \geq \theta$ or if the percentage of the set couples $\{(x_{ik}, x_{jl}) \mid sim(x_{ik}, x_{jl}) \geq \theta\}$ is greater than ϑ; where x_{ik} is a child node of x_i, and x_{jl} is a child node of x_j; θ and ϑ are two given thresholds.

Example 1. In Figure 3, considering $\theta = 0.7$ and $\vartheta = 0.7$, the following couples elements are highly similar: $sim(area, region) = 0.83 \geq \theta$; $sim(country, country) = 0.93 \geq \theta$; $sim(localName, vernacularName) = 0.78 \geq \theta$; $sim(reference, location) = 0.51 \leq \theta$ but $\{(x_{ik}, x_{jl}) \mid sim(x_{ik}, x_{jl}) \geq \theta\} = \frac{3}{4} = 0.75 \geq \vartheta$.

Definition 2. x_i and x_j are said to be structurally equivalent if they are highly similar and if the cardinal of the immediate children of x_i is equal to the cardinal of the immediate children of x_j.

Example 2. In Figure 3, (*ingredient*, *constituent*) are structurally equivalent.

Definition 3. x_i is said to be more general than y_j if they are highly similar and if the cardinal of the set of immediate children of x_i is greater than the cardinal of the set of immediate children of y_j.

Example 3. In Figure 3, the element name *reference* is more general than the element name *location*.

Definition 4. x_i is said to be structurally disjoint to y_j if they are not highly similar.

Example 4. In Figure 3, the element names *quantity* and *name* are structurally disjoint.

Definition 5. A tag matching $\mathtt{tag}(x_i, x_j)$ is a quadruple

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

where:

1. $sim(x_i, x_j)$ is the semantic similarity between x_i and x_j.
2. (a) $op(x_i, x_j) = \equiv$, if x_i and x_j are structurally equivalent;
   (b) $op(x_i, x_j) = \sqsubseteq$, if x_i or x_j is structurally more general;
   (c) $op(x_i, x_j) = \perp$, if x_i and x_j are structurally disjoints.

Example 5. In Figure 3: $\mathtt{tag}(constituent, ingredient) = (constituent, ingredient, 0.78, \equiv)$.

Theorem 1. Considering a set S of element names, the relation "highly similar" noted $\simeq$ is an equivalence relation on S.

Proof. Let x_i, x_j, x_k be three elements of the set S; c_i, c_j, c_k their respective concepts in a given taxonomy on the domain of application. Let θ, ϑ be two given thresholds. In the following proof, we are going to demonstrate the theorem using the first condition of highly similarity between elements that is $sim(x_i, x_j) \geq \theta$. As for the second condition: the percentage of the set couples $\{(x_{ik}, x_{jl}) \mid sim(x_{ik}, x_{jl}) \geq \theta\}$ is greater than ϑ; it can be deduced from the first condition when it is fulfilled for the set of child nodes elements.

1. *Reflexivity:* $x_i \simeq x_i$ because $IC(c_i) \equiv_1 IC(c_i)$; it follows that $sim(x_i, x_i) = 1 \geq \theta$ for any $\theta \in [0, 1]$, thus $\simeq$ is reflexive.
2. *Symmetry:* We assume $x_i \simeq x_j$ i.e. $sim(x_i, x_j) \geq \theta$. Let $sim(x_i, x_j) = \delta \geq \theta$, then $IC(c_i) \equiv_\delta IC(c_j)$ i.e. $IC(c_j) \equiv_\delta IC(c_i)$ thus $sim(x_j, x_i) = \delta \geq \theta$; and $\simeq$ is symmetric.
3. *Transitivity:* We assume $x_i \simeq x_j$ and $x_j \simeq x_k$ i.e. $sim(x_i, x_j) \geq \theta$ and $sim(x_j, x_k) \geq \theta$. Let $sim(x_i, x_j) = \delta_1$ and $sim(x_j, x_k) = \delta_2$ then $IC(c_i) \equiv_{\delta_1} IC(c_j)$ and $IC(c_j) \equiv_{\delta_2} IC(c_k)$; it follows from property $\mathcal{P}$ that $IC(c_i) \equiv_{\min(\delta_1, \delta_2)} IC(c_k)$, i.e. $sim(x_i, x_k) = \min(\delta_1, \delta_2) \geq \theta$ then $x_i \simeq x_k$ and $\simeq$ is transitive.

□

It follows from Theorem 1 that relation "highly similar", loosely speaking, partitions a set so that every element of the set is a member of one and only one cell of the partition. Two elements of the set are considered highly similar if and only if they are elements of the same cell. The intersection of any two cells is empty; the union of all the cells equals the original set. We therefore compute one-to-many matching by constructing groups of highly similar elements as deduced in Lemma 1.

Lemma 1. Let us consider m tag matching results of highly similar elements $\mathtt{tag}(x_1, x_2)\, \mathtt{tag}(x_2, x_3) \ldots \mathtt{tag}(x_m, x_{m+1})$. Then, there exists a tag matching result of

highly similar elements $\mathfrak{tag}(x_1, x_{m+1})$ and we can create a group of m highly similar elements

$$\mathfrak{grp}(x_1, \ldots, x_{m+1}) = (x_1, \ldots, x_{m+1}, sim(x_1, \ldots, x_{m+1}), op(x_1, \ldots, x_{m+1}))$$

where:

1. $sim(x_1, \ldots, x_{m+1}) = \frac{1}{m}\sum_{i=1}^{n} sim(x_i, x_{i+1})$
2. (a) $op(x_1, \ldots, x_{m+1}) = \equiv$, if $\forall (x_i, x_{i+1}), op(x_i, x_{i+1}) = \equiv$
   (b) $op(x_1, \ldots, x_{m+1}) = \sqsubseteq$, if $\exists (x_i, x_{i+1}) \,|\, op(x_i, x_{i+1}) = \sqsubseteq$.

From the group constructed, we marked some of them as structurally or semantically ambiguous using the following definitions.

Definition 6. A group of highly similar elements $\mathfrak{grp}(x_1, \ldots, x_{m+1})$ is said to be semantically ambiguous if $m + 1 \geq n$ or if there exist at least 2 elements x_i and x_j such that x_i and x_j both belong to the same source. n is the number of sources.

Example 6. In Figure 3,

$$\begin{aligned} \mathfrak{tag}(name, name) &= (name, name, 0.93, \sqsubseteq); \\ \mathfrak{tag}(name, communName) &= (name, communName, 0.75, \equiv) \\ \mathfrak{tag}(communName, scientificName) &= (communName, scientificName, 0.71, \equiv) \end{aligned}$$

then we can create $\mathfrak{grp}(name, name, scientificName, communName)$ where

$$\begin{aligned} sim(name, name, scientificName, communName) &= \frac{0.93 + 0.75 + 0.71}{3} = 0.79 \\ op(name, name, scientificName, communName) &= \sqsubseteq. \end{aligned}$$

$\mathfrak{grp}$ (*name*, *name*, *scientificName*, *communName*) is semantically ambiguous because its cardinal is greater than 2 (number of source) and also because *name*, *scientificName*, and *communName* belong to the same source schema.

Definition 7. A group of highly similar elements $\mathfrak{grp}(x_1, \ldots, x_{m+1})$ is said to be structurally ambiguous if there exist at least 2 tag matching results $\mathfrak{tag}(x_i, x_j)$, and $\mathfrak{tag}(x_k, x_l)$ such that $op(x_i, x_j) = op(x_k, x_l) = \sqsubseteq$.

Definition 8. A group of highly similar elements is ambiguous if it is semantically or structurally ambiguous.

3.3 Building Possible Mediated Schemas

Definition 9. Considering a set of n source schemas $\mathcal{S}_{i,i=1\ldots n}$ and their respective graph representations $\mathcal{G}_{i=1\ldots n} = (\mathcal{V}_i, \mathcal{A}_i), i = 1 \ldots n$, a possible mediated schema $\mathcal{T}$ computed from the set of sources $\mathcal{S}_{i,i=1\ldots n}$ is a directed graph $\mathcal{F} = (\mathcal{V}', \mathcal{A}')$ where $\mathcal{V}' \subset \cup_{i=1}^{n} \mathcal{V}_i$ and $\mathcal{A}' \subset \mathcal{V}' \times \mathcal{V}'$.

To provide the possible mediated schema, we first realize one-to-one matching between elements nodes names of two selected sources as showed in the function computeTag. The sources are selected using our predictive model called IHM (Information Hidden Model) presented in [6] which predicts data sources where a query result can be found based on three defined learning strategies: User Hidden Habit (UHH), User Hidden Background (UHB) and User Hidden keywords Semantics (UHS).

We first declare two new data types TagResults as a record and Graph as an adjacency list; that is an array list of adjacent elements.

```
01    Type TagResults
02    Begin
03        name1: string;
04        name2: string;
05        sim: real;
06        op = (equiv, moreGeneral, distinct);
07    End
```

```
01    Type Node
02    Begin
03        Var name: String; // the element node name
04        Var Node: ^nxt;
05    end;
06    Type List ^Node;
07    Type Graph: Array[1..n_b] of List; // n_b: number of nodes in a graph
```

The function computeTag takes two element names and their corresponding graphs and returns the tag matching result between the two elements names. It uses the functions sim and structSim to compute the semantic and structural similarity between two elements, respectively.

```
Function computeTag(x, y: string; G_i, G_j: Graph;): TagResults
01    Var tag: TagResults
02    Begin
03        tag.name1 ← x;
04        tag.name2 ← y;
05        tag.sim ← sim(x, y);
06        tag.op ← structSim(x, y);
07        return tag;
08    End
```

From the tag matching results, we select the highly similar element and we construct groups of highly similar elements as presented in the procedure computeGroup where the procedure insertTag adds a tag in a list; the procedure deleteTag deletes a tag from a list and the function compareTag compares two tags and returns true if the two tags are equivalent.

```
Procedure computeGroup(H: List)
01    Var
02        tag_1, tag_2: TagResults;
03        Grp_i: List;
04    Begin
05        i←1;
06        Repeat
07            tag_1 ← H.head;
08            Grp_i ← H.head;
09            deleteTag(tag_1, H);
10            tag_2 ←H.head;
11            For i=1 to |H| do
12                If compareTag(tag_1, tag_2)==true then
13                    insertTag(tag_2, Grp_i,);
14                    deleteTag(tag_2,H);
15                endIf
16            endFor
17            i ← i+1;
18        until |H|==0;
19    End
```

We further select a representative element from each group constructed using the function representElt. We then test the semantic and structural ambiguity of the different groups computed using the Boolean functions semAmbiguous and strAmbiguous, respectively. Therefore, from the semantically ambiguous groups of elements, we compute distinct subgroups using the function computeDistinctGroup; and we compute distinct nodes from the structurally ambiguous groups of elements using the function computeDistinctNode. The nodes computed are finally used to update the graph constructed using non-ambiguous groups of elements, as detailed in the procedure constructGraph. The graphs and nodes are constructed using the functions insertElt and insertNode which take an element and a node to be inserted and a graph as input, respectively and return the latter graph updated with the element or the node.

```
Procedure constructGraph(Grp_i: List;)
01    Var :
02        F_l, F_lj, F_lk: Graph;
03        G_ik: Node;
04        x_i, x_ij: String;
05    Begin
06        l ← 1;
07        If semAmbiguous(Grp_i)==false and strAmbiguous(Grp_i)==false then
08            x_i ← representElt(Grp_i);
09            F_l ← insertElt(x_i, F_l);
10            l ← l+1;
11        Else If semAmbiguous(Grp_i)==true then
```

```
12            Grp_ij ← computeDistinctGroup(Grp_i);
13            For each Grp_ij do
14                l_j ← l;
15                x_ij ← representElt(Grp_ij);
16                F_lj ← insertElt(x_ij, F_l);
17            endFor
18        Else If strAmbiguous(Grp_i)==true then
19            G_ik ← computeDistinctNode(Grp_i);
20            For each G_ik do
21                l_k ← l_j + 1;
22                F_lk ← select(F_lj);
23                F_lk ← insertNode (G_ik, F_lk);
24            endFor
25        endIf
26    End
```

We further select a third source and update the groups of highly similar elements by realizing one-to-one matching between elements of the new selected source and the representative elements from the group previously constructed. We repeat these steps until all the sources have been matched. Therefore, once a new source gets connected to the DSSP, the groups of highly similar elements are updated and new possible mediated schemas can be extracted. In order to deduce possible mediated schemas, we finally merge the graph constructed with the existing source schema. All the functions, data types and steps thereby described are used in Algorithm 1 to produce the possible mediated schemas.

Algorithm 1: Matching Algorithm

```
01    Input: Graph Representation of Source Schema G_1, G_2, ..., G_n
02    Output: Set of Graph Representation of possible mediated schema F_1, F_2, ..., F_m
03    Var
04        G_i, G_j, G_k: Graph;
05        x_i, x_j, x_k: string;
06        i,j,k,l,n,m: integer;
07        H, Grp_i: List;
08        θ: real;
09        tag, tag_1, tag_2:TagResult;
10    Begin
11        G_i ← selectGraph();
12        G_j ← selectGraph();
13        n ← 2;
14        For i=1 to |Gi| do
15            For j=1 to|Gj|do
16                tag ← computeTag(x_i, x_j, G_i, G_j);
17                if (tag.op ≠ distinct) and sim(x_i, x_j) ≥ θ then insertTag(tag, H);
18            endFor
19        endFor
20        computeGroup(H);
```

```
21      While G_k ← newSource() do
22          n ← n+1;
23          i ← 1;
24          Repeat
25              x_i ←representElt(Grp_i);
26              For k=1 to |Gk| do
27                  tag ← computeTag(x_i, x_k);
28                  if (tag.op ≠ distinct) and (sim(x_i, x_k) ≥ θ)
29                  then insertElt(x_k, Grp_i);
30              endFor
31              i ← i+1;
32          until i==m;
33      endWhile
34      For each Grp_i do constructGraph(Grp_i);
35      m ← 1;
36      For each F_lk do
37          F_m ← mergeGraph(F_lk);
38          m ← m+1;
39          Return F_m;
40      endFor
41  End.
```

The mediated schema constructed can be managed as if it is a single mediated schema based system. In other words, a possible mediated schema is constructed to behave as a whole mediated schema in itself. Therefore, when a source is selected to be queried, the system also selects one single mediated schema among the possible mediated schema and manages it as if it is a single mediated schema based system. Thus, to select a mediated schema related to the selected source, we propose to assign reliability degree to a given mediated schema with respect to the selected source. Thereby, instead of finding whether a mediated schema is consistent with a source, we compute its reliability degree (or degree of consistency) with respect to the source.

3.4 Reliable Mediated Schema (rMed)

Considering an instance $\mathcal{T}$ of the set of possible mediated schemas and a given source schema $\mathcal{S}_i$, our aim is to find out if $\mathcal{T}$ is reliable with $\mathcal{S}_i$ and assign to $\mathcal{T}$ a reliability degree with respect to $\mathcal{S}_i$.

3.4.1 Definitions

Definition 10. 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 $\mathcal{S}$.

To compute reliability degree of $\mathcal{T}$ with respect to $\mathcal{S}_i$, we first check if, in a structural point of view, $\mathcal{T}$ is reliable to $\mathcal{S}_i$ and we call it the *structural reliability degree.*

$\mathcal{T}$ is said to be structurally reliable with $\mathcal{S}_i$ 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.

The structural reliability degree is computed as the ratio between the numbers of sub-node of $\mathcal{T}$ that are structurally equivalent to a sub-node of $\mathcal{S}_i$ and the number of sub-node of $\mathcal{S}_i$. The structural reliability degree of $\mathcal{T}$ with respect to $\mathcal{S}_i$ noted $d^{Struct}_{\mathcal{T}/\mathcal{S}_i}$, is then written as follows:

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

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 Equation (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:

1. from Equation (1), $d^{Struct}_{\mathcal{T}/\mathcal{S}_i} \geq \alpha$
2. from Equation (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 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, $\sum_{e_i \in \mathcal{S}_i} d(e)\, p(e) \geq \theta \neq 0$ because $d(e)$ 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 schemas 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.

3.4.2 Example of Reliable Mediated Schemas

Let us consider 4 source schemas $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4$, and 3 possible mediated schemas M_1, M_2, M_3 in the field of African Traditional Medicine (ATM) as presented in Figure 4.

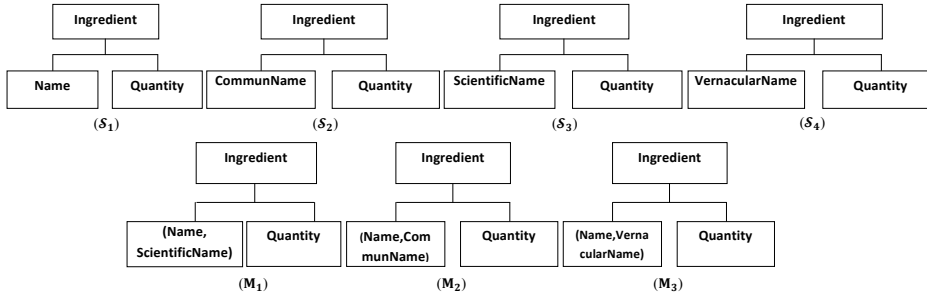

Fig. 4. Example of source schemas and possible mediated schemas

Considering the semantic similarity between highly similar elements of the 4 source schemas $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4$:

$$\begin{aligned} Sim(Ingredient, Ingredient) &= 0.91 \\ Sim(Quantity, Quantity) &= 0.93 \\ Sim(Name, ScientificName) &= 0.71 \\ Sim(Name, CommonName) &= 0.75 \\ Sim(Name, VernacularName) &= 0.72. \end{aligned}$$

In Table 1 we present the possible mediated schemas and their corresponding reliability degrees with respect to the source schemas: $(\mathbf{d}_{\mathbf{M_j}/\mathcal{S}_\mathbf{i}})$ and with respect to the whole set of data source schemas $(\mathbf{d}_{\mathbf{M_j}/\mathcal{S}_\mathbf{1}\ldots\mathcal{S}_\mathbf{4}})$. For example, using Equation (2):

$$d_{M_2/\mathcal{S}_2} = \frac{d(ingredient)p(Ingredient)+d(Quantity)p(Quantity)}{d(Ingredient)p(Ingredient)+d(Quantity)p(Quantity)+d(ScientificName)p(ScientificName)}$$

For example, we compute $p(ScientificName)$ and $d(ScientificName)$ as follows:

$$p(ScientificName) = \frac{number\ of\ mediated\ schema\ containing\ ScientificName}{Total\ Number\ of\ Mediated\ Schema} = \frac{1}{3}$$

$$d(ScientificName) = \frac{sim(Name, ScientificName) + sim(Name, CommunName) + sim(Name, VernacularName)}{3}$$

$$\begin{aligned} &= \frac{0.71 + 0.75 + 0.72}{3} \\ &= 0.72. \end{aligned}$$

Then, we have:

$$\begin{aligned} d_{M_2/\mathcal{S}_2} &= \frac{0.91\frac{3}{3} + 0.93\frac{3}{3}}{0.91\frac{3}{3} + 0.93\frac{3}{3} + 0.72\frac{1}{3}} \\ &= 0.88. \end{aligned}$$

	d_{M_j/S_1}	d_{M_j/S_2}	$d_{M_j/\mathcal{S}_3}$	$d_{M_j/\mathcal{S}_4}$	$d_{M_j/\mathcal{S}_1 \ldots \mathcal{S}_4}$
M_1	0.99	0.88	0.99	0.88	0.93
M_2	0.99	0.99	0.88	0.88	0.93
M_3	0.99	0.88	0.88	0.99	0.93

Table 1. Example of reliable mediated schema

Now, let us reconsider the 4 source schemas $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4$ showed on Figure 4 and the 2 mutually disjoint possible mediated schemas M_4, M_5 showed on Figure 5.

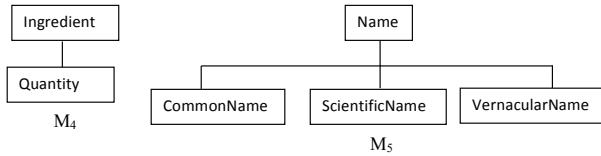

Fig. 5. Example 2: mutually disjoint possible mediated schemas

Table 2 presents the mutually disjoint possible mediated schemas M_4, M_5 and their corresponding reliability degrees with respect to the source schemas: $(\mathbf{d_{M_j/\mathcal{S}_i}})$ and with respect to the whole set of data source schemas $(\mathbf{d_{M_j/\mathcal{S}_1 \ldots \mathcal{S}_4}})$.

	d_{M_j/S_1}	d_{M_j/S_2}	$d_{M_j/\mathcal{S}_3}$	$d_{M_j/\mathcal{S}_4}$	$d_{M_j/\mathcal{S}_1 \ldots \mathcal{S}_4}$
M_4	0.717	0.717	0.717	0.717	0.717
M_5	0.282	0.282	0.282	0.282	0.282

Table 2. Example of mutually disjoint reliable mediated schemas

From these examples of possible mediated schemas we can observe that the reliability degree function behaves according to the set of possible mediated schema; and we can deduce the theorems presented in the following subsection.

3.4.3 Theorems

Theorem 2. Given a non-empty source $\mathcal{S}$ and m pairwise mutually disjoint mediated schemas $\mathcal{T}_{j,j=1\ldots m}$ such that $\mathcal{S}$ is a subset of $\mathcal{T}_1 \cup \mathcal{T}_2 \cup \cdots \cup \mathcal{T}_m$; we have $\sum_{j=1}^{m} d_{\mathcal{T}_j/\mathcal{S}} = 1$.

Proof.

$$\begin{aligned}\sum_{j=1}^{m} d_{\mathcal{T}_j/\mathcal{S}} &= d_{\mathcal{T}_1/\mathcal{S}} + d_{\mathcal{T}_2/\mathcal{S}} + \cdots + d_{\mathcal{T}_m/\mathcal{S}} \\ &= \frac{\sum_{e\in\mathcal{T}_1\cap\mathcal{S}} d(e)\, p(e) + \cdots + \sum_{e\in\mathcal{T}_m\cap\mathcal{S}} d(e)\, p(e)}{\sum_{e\in\mathcal{S}} d(e)\, p(e)}.\end{aligned}$$

The mediated schemas are pairwise mutually disjoint, this means that for $i \neq j$, $\mathcal{T}_i \cap \mathcal{T}_j = \emptyset$. It follows then that:

$$\begin{aligned}\sum_{e\in\mathcal{T}_1\cap\mathcal{S}} d(e)\, p(e) + \cdots + \sum_{e\in\mathcal{T}_m\cap\mathcal{S}} d(e)\, p(e) &= \sum_{e\in(\mathcal{T}_1\cap\mathcal{S})\cup\cdots\cup(\mathcal{T}_m\cap\mathcal{S})} d(e)\, p(e) \\ &= \sum_{e\in\mathcal{S}\cap(\mathcal{T}_1\cup\cdots\cup\mathcal{T}_m)} d(e)\, p(e) \\ &= \sum_{e\in\mathcal{S}} d(e)\, p(e).\end{aligned}$$

Therefore, $\sum_{j=1}^{m} d_{\mathcal{T}_j/\mathcal{S}} = 1$. □

Theorem 3. Given n nonempty sources $\mathcal{S}_{i,i=1\ldots n}$ and m pairwise mutually disjoint mediated schemas $\mathcal{T}_{j,j=1\ldots m}$ such that each $\mathcal{S}_i$ is a subset of $\mathcal{T}_1 \cup \mathcal{T}_2 \cup \cdots \cup \mathcal{T}_m$; we have $\sum_{j=1}^{m} d_{\mathcal{T}_j/\mathcal{S}_1,\mathcal{S}_2,\ldots,\mathcal{S}_n} = 1$.

The proof of Theorem 3 is immediate from Equation (3) and Theorem 2.

From Theorems 2 and 3, we can observe that the reliability degree function behaves according to the set of possible mediated schema. In other words, all depends on how the possible mediated schemas were constructed. If the mediated schemas were constructed to be pairwise mutually disjoint, the behavior of reliability degree function seems similar to the behavior of an unconditional probability distribution which depends on the number of possible mediated schemas. If the set of possible mediated schema is too large, as it usually occurs in DSSP, the reliability degree of some mediated schemas might be too low and the task of choosing a threshold might lead to information loss. It is then desirable to enhance the task of choosing thresholds. Therefore, constructing not mutually distinct possible mediated schemas could improve the value of reliability degrees. In this latter case, the reliability degrees seems to behave as a conditional probability distribution in which the probability of the intersections is contained in the value of the reliability degree.

We can finally address the next step of our method which is assigning reliability degrees to the possible mediated schemas with respect to the source schemas. In Algorithm 2 we present how reliability degrees are assigned to the possible mediated schema.

Algorithm 2: Reliable Mediated Schema

01 **Input:** Possible mediated schema $\mathcal{T}_1, \mathcal{T}_2 \ldots, \mathcal{T}_m$,
02 A Source Schema $\mathcal{S}_i$

03 **Output:** Reliable Mediated Schema $(\mathcal{T}_j, d_{\mathcal{T}_j/\mathcal{S}_i})$
04 Var α, β : real;
05 **Begin**
06 **For** j=1 **to** m **do**
07 **If** $(d^{Struct}_{\mathcal{T}_j/\mathcal{S}_i} \geq \alpha)$ **then**
08 Compute $d_{\mathcal{T}_j/\mathcal{S}_i}$;
09 **if** $d_{\mathcal{T}_j/\mathcal{S}_i} \geq \beta$ **then Return** $(\mathcal{T}_j, d_{\mathcal{T}_j/\mathcal{S}_i})$;
10 **EndIf**
11 **Endfor**
12 **End**

After computing reliable mediated schema our goal is to extract reliable semantic mappings (rMap) between a source schema and a target reliable mediated schema. Therefore, in the same way of thinking as presented previously, we first compute the tag matching between couple of elements names and we store in a group only the couples which are highly similar. From this group, we construct subgroups of highly similar elements and we mark the groups which are semantically ambiguous. Finally, from the non ambiguous group of elements, we return the mapping results, and from couple of ambiguous elements, we compute sets of possible mappings. Finally, we assign reliability degrees to the possible sets of mappings. We therefore formally define a reliable mapping as a couple $(\mathcal{M}, d(\mathcal{M}))$ where $\mathcal{M}$ is a set of distinct tag matching results between element names of the source schema and a target reliable mediated schema and $d(\mathcal{M})$ is the reliability degree of $\mathcal{M}$ with respect to the concerned source schema.

4 EXPERIMENTS

The algorithms presented above have been implemented and evaluated through a number of experiments. In this section we present the results obtained during these experiments. We especially check the feasibility, the efficiency, the accuracy and the performance of our system.

4.1 Experimental Setup

We build a reliable data integration system called KSpace based on the methods and algorithms presented in the previous sections. KSpace takes as input a set of data sources, internally represented as graph, and it automatically generates a set of possible mediated schemas. Kspace then assigns reliability to each mediated schema with respect to each of the sources and then deduces the reliability degrees of each mediated schema with respect to the set of data sources.

We used XML enabled Oracle database to store our data and we implement our methods, algorithms in C++. We used WordNet database [12] to compute Lin [14] similarity value between elements names. We conducted our experiments on a mixed network with three computers, one on Windows 7, another one on Linux-Fedora 10

and the last one on Ubuntu Desktop 9. Each computer used 2 CPUs Intel Pentium M 3 GHz with 1 Gb memory. For our experiments, we set the pairwise similarity threshold for creating the mediated schema to 0.70, and the structural threshold for creating mediated schema to 0.80.

We evaluate our system using standard metrics, Precision Recall and F-measure [23]. Considering a non-empty mapping $\mathcal{T}$ provided by a mapping tool and $\mathcal{T}_{ex}$ a non-empty mapping provided by a domain expert matcher:

Precision: expresses the proportion of correct mappings among the mappings produced by a mapping tool.

$$Precision = \frac{\mid \mathcal{T} \cap \mathcal{T}_{ex} \mid}{\mid \mathcal{T} \mid}.$$

Recall: shows the proportion of correct mappings extracted by the system, as a fraction of the expert.

$$\textbf{Recall} = \frac{\mid \mathcal{T} \cap \mathcal{T}_{ex} \mid}{\mid \mathcal{T}_{ex} \mid}.$$

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

$$F\text{-}measure = \frac{2 \cdot precision \cdot recall}{precision + recall}.$$

4.2 Experimental Results

We first evaluate the feasibility of our system using real world data sources in African Traditional Medicine (ATM) domain on 3 scenarios plants, diseases and treatments. The sources were selected from different projects [7, 25, 24, 26] on ATM from diverse African countries. Each sub-domain, plant, disease or treatment contains hundreds of documents translated into French. We compute the average obtained with each mediated schema (rMed) and reliable semantic mappings (rMap) on precision, recall and F-measure as showed in Figure 6.

After that, we evaluate the efficiency of our system by observing the response time of our system on the number of input sources as showed in Figure 7.

Then, we evaluate the accuracy of our system by comparing the accuracy of the mediated schema generated with the mediated schema generated using semi-automatic system. We therefore used XBenchMatch [5], a benchmark for XML schema matching tool, to compare the accuracy of reliable mediated schemas and then compared our system (KSpace) to COMA, Similarity Flooding and PORSCHE on precision, recall and F-measure for 3 scenarios: person, university and biology (Figure 8).

We finally evaluate the performance of our system by comparing the response time between KSpace and P-mapping system (referred to here as UDI) as shown in Figure 9.

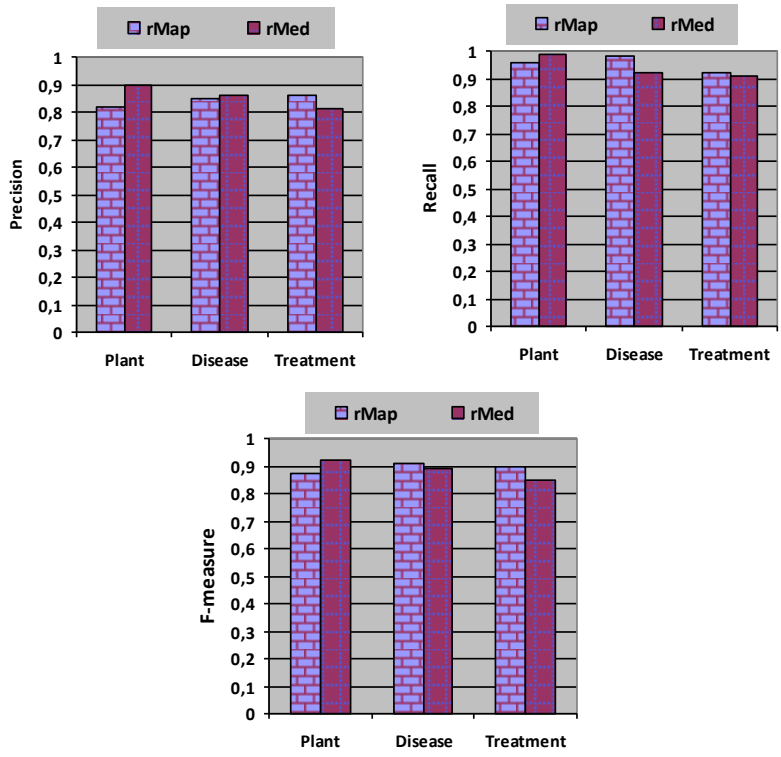

Fig. 6. Precion-Recall-F-measure of KSpace on 3 scenarios (Plant, Disease, and Treatment): The standard metrics precision, recall and F-measure of rMed and rMap vary above 0.8

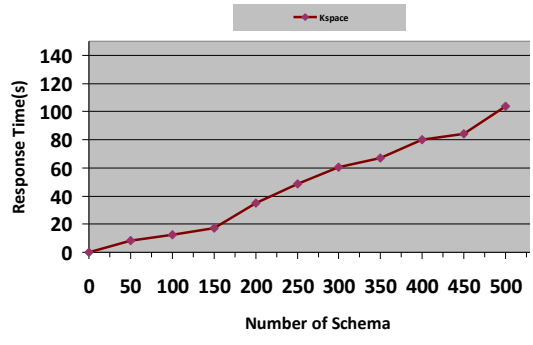

Fig. 7. KSpace response time on the number of input schema (Scenario Plant): The response time is a linear function on the number of input sources

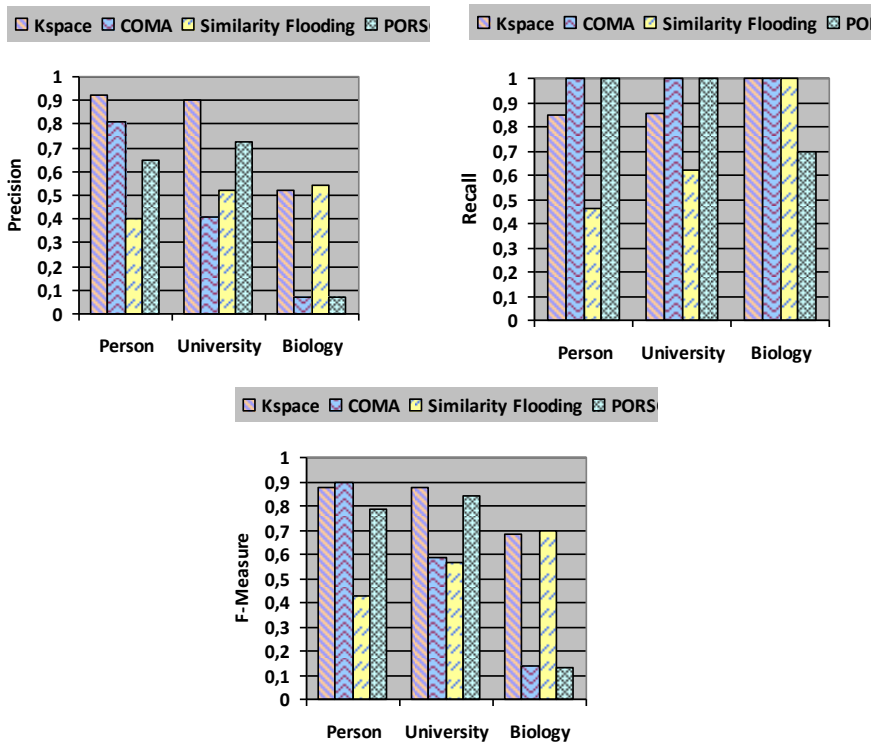

Fig. 8. Precion-Recall-F-measure of KSpace compared with COMA, Similarity Flooding, PORSCHE on 3 scenarios Person, University and Biology

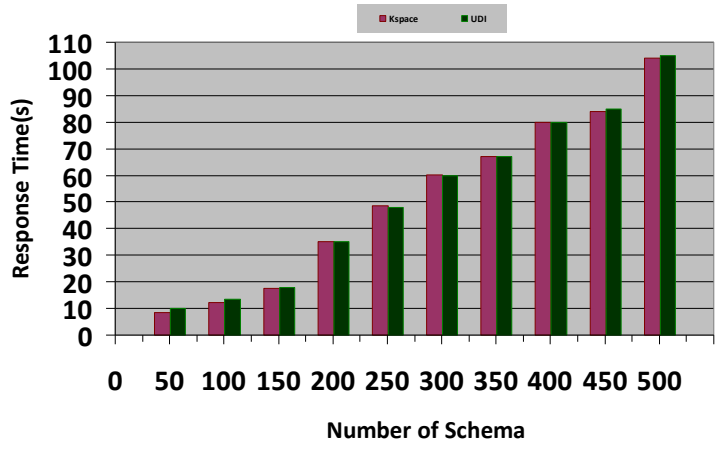

Fig. 9. KSpace(rMedMap) compared with UDI(pmapping)

As we can deduce from Figure 8, although our system is a self-sustained based system, it produces reliable mediated schemas which seem as accurate as the single mediated schema produced by semi-automatic systems. As for the response time, Figure 9 shows that KSapce is faster than UDI for less than 150 input schemas and above 150 input schemas, KSpace evolves sometimes a second more than UDI.

5 RELATED WORK

In this section we present some existing methods which also perform matching and mapping between source schemas in order to generate mediated schemas and semantic mappings. We argue about the necessity to propose new solutions for data integration in Dataspace Support Platforms which are self-sustained based systems. The reader can refer to [17] for a survey on schema matching approaches or to [3, 16, 20, 15, 13, 4, 9, 2, 8] for recent methods on schema mappings.

PORSCHE [20] provides single mediated schema from a set of multiple tree-based input schemas. Their method utilizes a holistic approach which first clusters the nodes based on linguistic label similarity. Similarity Flooding [16] proposes a matching algorithm based on a fixpoint computation that is usable across different scenarios. It takes two graphs schemas as input, and produces as output a mapping between corresponding nodes of the schemas. After their algorithm runs, they expect a human to check and if necessary adjust the results. Unfortunately, in the possible application domain of dataspace, the user might not be skilled enough to manipulate mappers or to provide accurate mappings.

COMA [3] provides semantic mappings between two input schemas by combining multiple matchers presented in the survey [17] and represents a generic match system supporting different applications and multiple schema types such as XML and relational schemas. In the same path, He and Chang [13] assume that the source schemas are created by a generative model applied to some mediated schema. Our approach does not depend on a specific matchers or a particular schema-matching technique, when these approaches are generic models and thus must rely on statistical properties of source schemas.

In [15] the authors proposed a method generating a set of alternative mediated schemas based on probabilistic relationships between existing integrated relations. In a DSSP, sources might not be fully integrated before use. Then we can not focus on the properties generally obtained when we fully integrated source schemas. That is the reason why we focus on the matching between element nodes names to lead the system self-extracts useful relationships between its participants. Our system also reuses the self-extracted relationship and therefore improves the type of relationships as a new source gets connected in a pay-as-you-go [11] fashion. All the contradictions presented above arise especially because a DSSP is a self-sustained based system and it has special properties different from those of existing data integration systems. Therefore, new solutions are needed in the case of data integration in DSSP.

Dong et al. [9] proposed the concept of probabilistic schema mapping for data integration with uncertainty in DSSP, but they did not describe how to create such mappings. Sarma et al. further introduced p-mapping [2], a probabilistic-based method to manage uncertainty between the semantic mappings automatically generated between a source schema and a target mediated schema. Assigning of probabilities is obtained by computing the ratio between the number of source a given mediated schemas is consistent with and the total number of source. Then to compute the numerator of this fraction, the system should determine whether a given mediated schema is consistent with a source. Hence, the consistency measurement of a given mediated schema or semantic mapping with respect to a source is a value in the set $\{0, 1\}$. Therefore, the mapping or mediated schema with the highest probability will be the one which is consistent with the highest amount of sources; but, a given mediated schema could be consistent with a low number of sources, therefore its probability will be too low and a system constructed on such type of measure might not consider the information available in those sources; this might lead to information loss. We then propose to fuzzify the consistency measurement by computing the "*degree of consistency*" or "*reliability degree*" which belongs to the interval [0,1] instead of belonging to the set $\{0, 1\}$. This will then lead the system to consider as much as possible the information available in the source in order to produce "*best endeavor results*". We think using such a technique will lead to a better management of uncertainty among the mediated schema or semantic mappings self-provided.

6 CONCLUSION AND FUTURE WORKS

In this paper we present the Reliable Matching and Mapping method (rMedMap) which automatically provides reliable mediated schemas (rMed) and reliable mappings (rMap) from a set of independently constructed source schemas in DataSpace Support Platforms which is a self-sustained system. Our aim was to increase the system's endeavor results by leading it to considering as much as possible information contained in any source connected. We then present algorithms which automatically provide set of possible mediated schema. We further show how to compute reliability degrees and assign to the possible mediated schema with respect to each source schema and to the whole set of data source schemas connected to the DSSP. Compared to existing systems, experimental results show that our system is faster and, although completely automatic, it produces reliable mediated schemas which are as accurate as those produced by semi-automatic systems. Moreover, with less than a hundred and fifty input schemas, the response time of our system is less than the response time of the unique existing system which can also produce multiple probabilistic mediated schemas for data integration in DSSP. Finally, our aim seems to be achieved because we led the system considered as much as possible information contained in the source in order to produce its best endeavor results.

As for the future work, we discover that providing reliability degrees to manage uncertainty between mediated schemas and semantic mappings in Dataspace systems is necessary but not sufficient; because there exists another level of uncertainty between reliable mappings which has to be handled. In other words, in some cases the quality of a reliability degree or probability distribution might depend on the number of possible mappings. That is, if the set of possible mappings is too large, the reliability degrees or the probability distribution will be too low. We are then planning to use possibility theory to manage such uncertainty between the reliable mappings.

Acknowledgment

This work is supported by Jiangsu Provincial Natural Science Foundation of China (No. BK2011454), the Beijing Municipal Commission of Education (Grant No. KM-201111417002) and the Open Foundation from Computer Application Technology in the Most Important Subjects of Zhejiang (No. 10Y0001).

Our thanks to all the projects on African Traditional Medicine who were willing to put their database available to us during the experiments.

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

REFERENCES

[1] Couto, F.—Silva, M.—Coutinho, P.: Measuring Semantic Similarity Between Gene Ontology Terms. Data Knowledge Engineering, Vol. 61, 2007, No. 1, pp. 137–152.

[2] Das Sarma, A.—Dong, L.—Halevy, A.: Bootstrapping Pay-as-You-Go Data Integration Systems. Proc. of the 28th ACM SIGMOD, Vancouver, Canada, June 9–12, 2008.

[3] Do, H.-H.—Rahm, E.: Matching Large Schemas: Approaches and Evaluation. Journal of Information Systems, Vol. 32, 2007, No. 6, pp. 857–885.

[4] Doan, A.—Madhavan, J.—Domingos, P.—Halevy, P. Y.: Learning to Map Between Ontologies on the SemanticWeb. Proc. of the International World Wide Web Conference (WWW) 2002.

[5] Duchateau, F.—Bellahsene, Z.—Hunt, E.: XBenchMatch: A Benchmark for XML Schema Matching Tools. Proc. of the 33rd International Conference on Very Large Data Bases (VLDB), Vienna, Austria 2007.

[6] Fanzou Tchuissang, G. N.—Wang, N.—Kuicheu, N. C.—Siewe, F.—Liu, S.—Xu, D.: Predicting DataSpace Retrieval Using Probalistic Hidden Information. IEICE Transaction on Information and Systems, Vol. E93-D, 2010, No. 7.

[7] Fotso, L. P.: Table of Entities and Attributes of Data Bases in MEDITRA. Rapport de recherche, No. 20, University of Yaounde 1, February 1999 (in French).

[8] Dong, X.: Providing Best-Efforts Services in Dataspace Systems. Ph. D. Dissertation, University of Washington 2007.

[9] Dong, X.—Halevy, A. Y.—Yu, C.: Data Integration with Uncertainty. Proc. of International Conference on Very Large DataBases (VLDB) 2007.

[10] Franklin, B.—Halevy, A.—Maier, D.: From Databases to Dataspaces: A New Abstraction for Information Management. ACM SIGMOD Record, Vol. 34, Issue 4, pp. 27–33, December 2005.

[11] Halevy, A.—Franklin, M.—Maier, D.: Principles of Dataspace Systems. Proc. 25th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS). Chicago, USA 2006, pp. 1–9.

[12] Harabagiu, S. M.—Miller, G. A.—Moldova, D. I.: WordNet 2 – A Morphologically and Semantically Enhanced Resource. Proc. of the ACL SIGLEX Workshop: Standardizing Lexical Resources, University of Maryland 1999.

[13] He, B.—Chang, K. C.: Statistical Schema Matching Across web Query Interfaces. Proc. of ACM SIGMOD 2003.

[14] Lin, D.: An Information-Theoretic Definition of Similarity. Proc. of the 15th International Conference on Machine Learning (ICML) 1998, pp. 296–304.

[15] Magnani, M.—Montesi, D.: Uncertainty in Data Integration: Current Approaches and Open Problems. Proc. of International Conference on Very Large DataBases (VLDB), Workshop on Management of Uncertain Data 2007.

[16] Melnik, S.—Garcia-Molina, H.—Rahm, E.: Similarity Flooding: A Versatile Graph Matching Algorithm and its Application to Schema Matching. Proc. of the 18th IEEE International Conference on Data Engineering (ICDE), San Jose, CA 2002.

[17] Rahm, E.—Bernstein, P. A.: A Survey of Approaches to Automatic Schema Matching. VLDB Journal, Vol. 10, 2001, No. 4, pp. 334–350.

[18] Resnik, P.: Semantic Similarity in a Taxonomy: An Information-Based Measure and Its Application to Problems of Ambiguity in Natural Language. Journal of Artificial Intelligence Research, Vol. 11, 1999.

[19] Ross, S. M.: A First Course in Probability. Prentice Hall 1998.

[20] Saleem, K.—Bellahsene, Z.: PORSCHE: Performance ORiented SCHEma Matching. Proc. of the 16th ACM International World Wide Web Conference (WWW), Banff, Canada, 2007.

[21] Schlicker, A.—Domingues, F. S.—Rahnenfuhrer, J.—Lengauer, T.: A New Measure for Functional Similarity of Gene Products Based on Gene Ontology. BMC Bioinformatics, Vol. 7, 2006, No. 1, p. 302.

[22] Schwering, A.: Approaches to Semantic Similarity Measurement for Geo-Spatial Data: A Survey. Transactions in GIS, Vol. 12, 2008, No. 1.

[23] Van-Risbergen, C.: Information Retrieval. 2nd edition, Butterworths, London 1979.

[24] AFRITRADOMEDIC: `http://www.afritradomedic.com`.

[25] MetAfro: `http://www.metafro.be`.

[26] MRC: `http://www.mrc.ac.za`.

Nathalie Cindy Kuicheu graduated from the University of Dschang, Cameroon in 2003, and went to The University of Yaound 1, Cameroon where she got her Maitrise in computer science in 2004 and defended her Master Degree with thesis in June 2006. She then gained a China-Cameroon Government Scholarship and went to China to pursue her studies where she is currently a Ph. D. candidate at Beijing Jiaotong University. Her research interests include data integration, semantic information management, iconic communication and African traditional medicine information management.

Ning Wang received her Ph. D. degree in computer science in 1998 from Southeast University in Nanjing, China. She is now an Associate Professor of computer science in Beijing Jiaotong University. Her research areas include web data integration, web search, XML keyword search, personal information management.

Gile Narcisse Fanzou Tchuissang graduated in 2002 from the University of Dschang, Cameroon where he received his Bachelor Degree. He continued his studies at the University of Yaoundé I where he finished his Maitrise and defended his Master Degree with thesis in March 2006. He obtained a UNESCO scholarship and went to China to carry out some research at Beijing Jiaotong University, where he recently received his Ph. D. degree in computer science. His research interests include information retrieval, keyword search, image search, XML information management.

De Xu received his Bachelor in applied mathematics from University of Science and Technology of China in 1967. He received the Master from Beijing Jiaotong University in 1982 and is currently Professor at the School of Computer and Information Technology. During all these years, he went to many universities in North America and Asia-Pacific as Visiting Scholar and Visiting Professor. All his experience includes hundreds of papers, books and tens of students supervised. His research interests include visual cognitive computing and image processing.

Guojun Dai received his B. E. and M. E. degrees from Zhejiang University in 1988 and 1991, respectively, and the Ph. D. degree from the College of Electrical Engineering, Zhejiang University in 1998. He is currently a Professor and the Deputy Dean of the College of Computer Science, Hangzhou Dianzi University, Hangzhou, China. He is the author or co-author of more than 20 papers and books in recent years, and holds more than 10 patents. His research interests include biomedical signal processing, computer vision, embedded systems design, and wireless sensor networks.

François Siewe is currently a senior research fellow at De Montfort University (DMU) in Leicester, United Kingdom. He received his Ph. D. in computer science from the Software Technology Research Laboratory at DMU in 2005. He obtained his B. Sc. in mathematics and computer science, M. Sc. and Doctorat de Troisième Cycle in computer science from the University of Yaounde I, Cameroon in 1990, 1992 and 1997, respectively. His research interests include computer security, context-aware systems, ubiquitous and pervasive computing, and formal verification.