A Flexible Approach Based on the user Preferences for Schema Matching

Wided Guédria
Faculté des Sciences de Tunis,
Campus universitaire,
1060 Tunis, Tunisie,
Email: wided.guedria@lirmm.fr

Zohra Bellahsène
LIRMM - UMR 5506,
Université Montpellier 2,
34392 Montpellier Cedex 5 - France
Email: bella@lirmm.fr

Mathieu Roche
LIRMM - UMR 5506,
Université Montpellier 2,
34392 Montpellier Cedex 5 - France
Email: mroche@lirmm.fr

***Abstract*— Automating schema matching is challenging. Previous approaches focus on computing all element matches between two schemas and don't take into account the preferences of the user who can be only interested in specific elements of the schema. We propose a new approach based on the user preferences to extract subsets of schemas on which will be applied the matching process. Fuzzy sets can be used to express the user preferences in the selection criteria of a query. Thus, we introduce the notion of fuzzy set defined over a part of the schema, then its extended form that is explicitly defined over the whole schema, according to the generalization rules. This will reduce the research space and therefore contribute to optimize the schema matching process. We also propose to propagate weights to elements of a target schema according to the user preferences on a source schema and mappings found by the matcher between the two schemas. The output scores give an automatic order of the target schema elements based on the interest expressed by the user.**

Keywords: Schema matching, mappings, user preferences, similarity degree, fuzzy subset on schema, preferences propagation.

I. Introduction

Schema matching is the task of finding semantic correspondences between elements of two schemas. This is the main issue in many database application domains, such as heterogeneous database integration [24], E-commerce, data warehousing and semantic query processing [21].
Numerous systems and approaches have recently been developed to determine schema matches semi-automatically {[1], [10], [12], [15], [16], [17], [18], [20]}.
Given two schemas, the output of most matching systems is a set of semantic correspondences (or mappings) between attributes of schemas (see figure 1 [3]).
Most of schema matching approaches have emerged from the context of a specific application. Only few approaches (Clio [17], COMA [10], Cupid [15], and SF [16]), try to address the schema matching problem in a generic way which is suitable for different applications and schema languages. In the following, we present an overview of two approches (LSD and GLUE) achieved for specific applications and two others (COMA and COMA++) designed in a generic way.

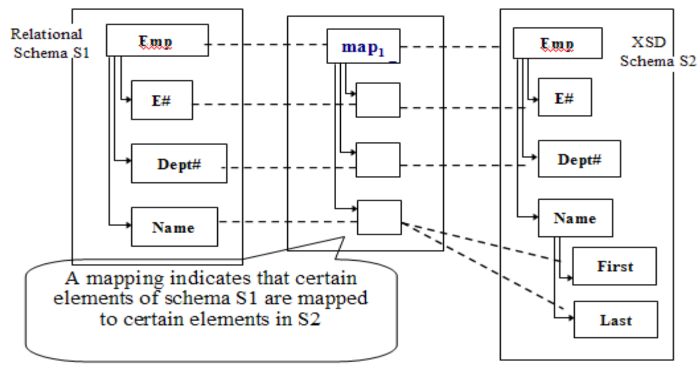


FIG. 1 –. *Example of matching between two schemas.*

– LSD [8] and its extension GLUE [9] use a composite approach to combining different matchers. They employ and extend current machine-learning techniques to semi-automatically find mappings. They were developed mainly for the XML[1] domain. LSD first asks the user to provide the semantic mappings for a small set of data sources, then uses these mappings together with the sources to train a set of learners. While LSD matches new data sources to a previously determined global schema, GLUE deals with ontologies and performs matching directly between the data sources. Both use machine-learning techniques for individual matchers and an automatic combination of match results.
– COMA [10] and its extension COMA++ [1] were developed for combining match algorithms in a flexible way. They represent generic match systems supporting different applications and multiple schema types such as XML and relational schemas. They follow a composite approach, which provides an extensible library of different matchers and support various ways for combining match results. COMA and COMA++ reuse previously obtained match results which may lead to significant savings of manual effort. Moreover, the two approaches are used as an evaluation platform to systemat-

. Supported by ANR Research Grant ANR-05-MMSA-0007

1. eXtensible Markup Language

ically examine and compare the effectiveness of different matchers and combination strategies. COMA++ extends the COMA prototype with major improvements like a comprehensive graphical user interface, a generic data model to uniformly support schemas and ontologies and a variety of high-level operators to compose, merge or compare different mappings.

Motivations

To reduce the user effort as much as possible, a substantial research and development effort have been achieved in order to provide him with semi-automatic solutions. Many systems have been developed and several of them evaluated to show their effectiveness [20]. However, the related work don't take into account user preferences and suppose that the whole schema interests the user whereas he can be only interested in specific elements.
Moreover, schemas to match can have an important size. Thus matching operation may entail high time consuming cost while time measure is an important and valuable part of schema matchers evaluation.
To solve these problems, it is important to know the interest that the user carries to elements of a schema and to let him express his preferences through a query. For that, we were influenced by [6] to propose a new approach. From preference degrees assigned to elements of a source schema, we form subsets on schema (preference classes). This will reduce the research space, therefore, it contributes to the optimization of the schema matching process. Our approach includes also a propagation of degrees assigned to elements from the source schema to the target ones.
Weights are assigned to elements of a target schema according to the user preferences on a source schema and mappings found by the matcher between the two schemas.

Thus, the main contributions of this paper are:

– The definition of the ***fuzzy subset schema***, that may be over a part of a schema and its ***generalization***, that is explicitly defined over the whole schema, using the links between elements of the schema. This will reduce the research space of the schema matching operation.
– The ***propagation*** of the user preference degrees to elements of a target schema according to his preferences on a source schema and mappings found between the two schemas. Fuzzy relations are used as a basis to define the propagated preferences degrees.

The outline of the paper is as follows: We begin in Section II "preliminaries" with definitions of the important concepts used in our research context.
In section III "Related work", we present a brief overview of some studies that are closed to our approach.
In section IV "Fuzzy subset schema approach", we introduce the notion of fuzzy set defined over a subset of a schema, then we present its extended form defined over the whole schema. We describe the propagation process of the user preferences from a source schema toward target ones and we give an example.
In section V "Experiments and evaluations", we present some results from the application of the presented method.
Finally in section VI "Summary and future work", we list our concluding remarks and future work.

II. PRELIMINARIES

We present in this part definitions of the important concepts used in our research context.

1) ***Similarity measure*** is a concept whereby two or more terms are assigned a metric value: Similarity degree, in the range of $[0,1]$ based on the likeness of their meaning / semantic content.
2) A **schema** is a labeled unordered tree [14] *S = (*V_S, E_S, r_S*, label)* with: V_S is a set of nodes, r_S is the root node, $E_S \subseteq V_S \times V_S$ is a set of edges, *label* $V_S \rightarrow$ Ł where Ł is a countable set of labels.
3) ***Schema Matching*** is the discovery of mappings between related schema elements belonging to disparate data sources. It takes two schemas as input and produces a semantic correspondence between the schema elements in the two input schemas.
4) In its simplest form, a ***mapping*** is a set of element matches each of which binds a source schema element to a target schema element if the two schema elements are semantically equivalent.

III. RELATED WORK

In real-world applications, information is often imperfect. So fuzzy set theory has been applied in a number of real applications crossing over a broad realm of domains and disciplines $\{$[2], [6], [13], [22], [23], etc.$\}$. We present in this part related work based on the use of fuzzy sets and that are closed to our approach.

A. Fuzzy sets and ontologies

Whereas in classic fuzzy sets, all the elements are on the same level and are associated with a degree explicitly defined, this is not necessarily the case in hierarchical fuzzy sets because several levels of detail exist in the hierarchy, and the hierarchical links between the elements have to be taken into account.
In [6], the hierarchical links are defined by the "kind of" relation. Such a domain is called an ontology. The membership of an element in a fuzzy set has consequences on the membership of its sub-elements. The approach presents the notion of fuzzy set defined over a subset of the ontology, then its developed form defined over the whole ontology.
This method has been applied within the information system of the SymPrevius project, which brings together industrial and academic partners to build a tool for the analysis of microbiological risks in food products (http://www.symprevius.org). The fuzzy set formalism was used in two main ways:

– In the data modeling, for representing imprecise data expressed in terms of possibility distributions.

– In the query expression, for representing fuzzy selection criteria which express the preferences of the user.

B. Fuzzy set approach in Information retrieval

Recently, numerous Information Retrieval (IR) models have been designed based on concepts rather than keywords. The concept-based Information Retrieval aims at retrieving relevant documents on the basis of their meaning rather than their keywords. The main idea at the basis of conceptual IR, is that the meaning of a text depends on conceptual relationships between real world objects rather than linguistic relations found in the text or dictionaries.
In [2], the proposed approach is based on the use of a fuzzy conceptual structure both to index document and to express user queries. The documents are represented as a hierarchy like ontology where nodes are weighted. As a consequence, also queries are based on weighted keywords and presented as a weigted tree. The query evaluation is based on the comparison of minimal subtrees containing the two sets of nodes corresponding to the concepts expressed respectively in the document and the query.
Fuzzy operators are used in this comparison to avoid the rigidity that a classical comparison could give. Indeed, here even though nodes of the two subtrees are not identical, a degree of matching is calculated taking their possible common parents into account.

IV. Fuzzy subset schema approach

Current schema matching systems don't take into account the user preferences. Moreover, the time taken by the system to produce mappings between schemas is not going to have the same importance for large schemas and small ones. To solve these problems, we propose an approach based on the preferences of the user to extract subsets of schemas on which will be applied the matching process.

A. Fuzzy subsets over schemas

A "Fuzzy Subset over a Schema" (that we will note FSS) is a subset of elements on a schema S, where, to each element of the schema is assigned a user preference degree.
In a query, the user will associate to two elements e_1 and e_2 of the schema his preference degrees. e_1 and e_2 are considered as keywords. He can affect a degree d_1 to e_1 and a degree d_2 to e_2, where for example, $d_2 \leq d_1$, with, e_1 is a predecessor of e_2 on the graph representing the schema. It's from this query which contains the user preferences that the fuzzy subsets will be formed.
Preference degrees can be determined semi-automatically; in this case, the user have to give an order of preferences and the system will assign degrees according to this order.

B. Example 1

Let's consider the schema PO of the figure 2 that describes purchase orders with its lines (POLines), invoices (PoBillTo) and deliveries (PoShipTo) [15] and the FSS $\{0{,}6$/POLines, $0{,}7$/ POBillTo$\}$ including POLines and POBillTo with the respective preference degrees $0{,}6$ and $0{,}7$. Preference degrees assigned to elements of the PO schema are schown in figure 2.

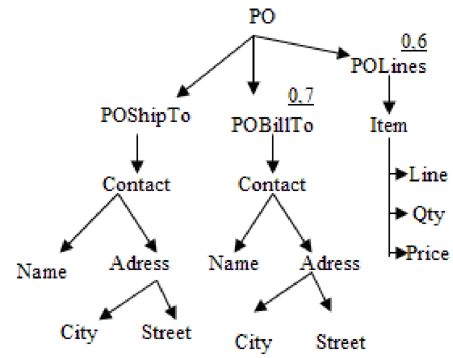


FIG. 2 –. *Presentation of the FSS* $\{0{,}6$/*POLines*, $0{,}7$/ *POBillTo*$\}$.

As it is illustrated in this example, the resulting fuzzy subset schemas are defined over two different parts of the schema and not on whole the schema, what prevents to use the classic comparisons between fuzzy subsets to compare the FSS.

C. Generalization of the "Fuzzy subsets over schemas"

The main objective of the FSS generalization is to discover concepts whose user can need and that could have omitted.
For example, if in a preference expression, the user specifies the schema element Item, we consider that he is interested naturally to everything relating to Item. On the other hand, we consider that a predecessor of an element in a schema is too general to be pertinent.
In the following, we define generalization rules of the FSS which were inspired from [6].

Generalization rules*: Let S be a schema and s_1 an FSS over dom(s1) (where dom(s_1) $\subseteq$ dom(S)) with a membership function μ_{s_1}.*
For all element e of the global schema S, let $Pred(e) = \{e_1$, ..., $e_n\}$ be the set of predecessors in the schema structure. The generalization of a FSS noted ext(s1), is defined over the whole schema S and is achieved according to the following rules:

1) *If e is in the FSS, then e preserves the same degree in its generalization.*
2) *If e has a unique predecessor e_1 in the FSS, then the degree of e_1 is propagated to e in the generalization.*
3) *If e has several predecessors $\{e_1$, ..., $e_n\}$ in the FSS with different degrees, a choice must be established concerning the degree that will be affected to e in the generalization. The proposed choice is to take the maximum degree of e_1, ..., e_n, since the user is interested to specific concepts (successors) and not to generalities.*
4) *All other elements, such those not descended from the starting FSS, generalizations, and no comparable elements with those in the FSS, are considered as no pertinent, the degree 0 is associated to them.*

From the generalization of the FSS, classes of preferences will be formed according to degrees affected to schema elements. A class of preferences will contain all nodes belonging to the same hierarchy in a schema and having the same preference degree.

D. Example 2

Let's consider the corresponding FSS built from the query user $\{0{,}6$ /POLines, $0{,}7$ /POBillTo$\}$. Degrees of preference affected to the concepts POLines and PoBillTo will be propagated to the other concepts according to generalization rules specified above and preference classes will be formed. As we can see in figure 3, two classes are resulting from this generalization.

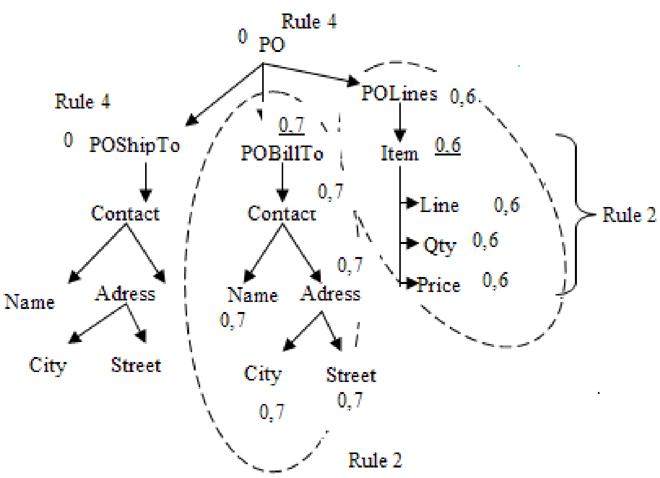


FIG. 3 –. *Preference classes related to the FSS* $\{0{,}6$ */POLines,* $0{,}7$ */ POBillTo*$\}$.

E. Matching of the "Fuzzy sets on schemas"

Once preference classes are formed, we will apply the matching process between those classes and a target schema. For that, we have used the COMA++ system for several reasons. In [20], the evaluations of different match prototypes, was performed. COMA seems to be quite successful.
In [1], COMA++ has shown much faster execution times and better results than COMA, especially in large match problems. Moreover, COMA++ have a graphical interface enabling a variety of user interactions and allow the application of different match strategies. within these strategies, we can mention:

- AllContext: Context-dependent match strategy. It allows matching all contexts of input schemas by determining all paths from the schema root to a node.
- FilteredContext: Refinement-based strategy for context-dependent matching. It identifies first similar nodes, then match the contexts of the similar nodes.
- Reuse: This strategy determines mapping paths from existing match results to solve a new match tasks.

In our approach, we use the "AllContext" strategy since it gives all existing mappings between two schemas. COMA++ provides correspondences between schema elements with similarity degrees. The similarity value is between 0 and 1. Thus, our matching relation is applied on fuzzy subsets (FSS) and it gives couples of elements with similarity degrees between 0 and 1. We propose to modelize it as a fuzzy relation [25].
A fuzzy relation R between two sets X and Y is a fuzzy subset defined over the universe $U1 \times U2$ with membership function μ_R as:

$$\mu_R\colon U1 \times U2 \to [0{,}1]$$
$$(x\ ,y) \mapsto \mu_R(x{,}y).$$

Given a fuzzy relation, we can use the compositional rule which represents the inference rule in fuzzy logic [25]. This allows us to find preferences of the use over the target schema from his preferences over the source one.

***Definition**:* Let I be a set of input values, O a set of output values and E a knowledge on I.
The compositional rule deals with the following issue: Given a knowledge E and a fuzzy relation R between I and O, what are the values that can take the output? [19].
The mechanism of inference is schematized by:

$$\begin{array}{c} R \in F(U \times V) \\ E \in F(U) \\ \hline ?\ F \in F(V) \end{array}$$

with $\mu_F(v)_{v \in V} = max_{u \in U}[min(\ \mu_E(u),\ \mu_R(u{,}v))]$ and " F $\in F(V)$ " represents the output to determine {[19], [27]}.

F. Example 3

We will present in this example an application of the compositional rule and show how the degrees affected to elements of the source schema PO (figure 2) can be propagated to the target schema (see figure 4). For that, we will match the preference classes (fragments) of the schema PO and a target schema which represents also purchase orders (figure 4) [15].

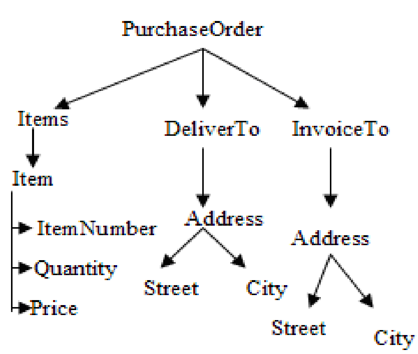


FIG. 4 –. *The PurchaseOrder schema.*

According to the generalization rules, the user preferences on the source schema PO are (see figure 3):
$\{$(POLines)(0,6), (Item)(0,6), (Line)(0,6), (Qty)(0,6), (Price)(0,6), (POBillTo)(0,7), (Contact)(0,7), (Name)(0,7), (Adress)(0,7), (Street)(0,7), (City)(0,7)$\}$.

We have used COMA++ to find mappings between the two schemas (figure 2 and figure 4).
The resulting correspondences are represented by the following relation:

R = "maps to" = {(POLines, Items)(0,57), (Item, Item)(0,72), (Line, ItemNumber)(0,65), (Qty, Quantity)(0,75), (Price, Price) (0,75), (POBillTo, InvoiceTo) (0,62), (Adress, Adress) (0,51), (Street, Street) (0,76), (City, City) (0,76)}.

This relation contains pairs of elements with their respective similarity degrees. For example POLines (in the source schema PO) corresponds to items (in the target schema PurchaseOrder) with the similarity degree 0,57.

Given this fuzzy relation and the user preferences on the source schema PO, we apply the compositional rule to propagate these degrees on the target schema PurchaseOrder.
For example we will compute the degree propagated to the items. According to the compositional rule, this degree is equal to the maximum of the set of minimums similarity values between elements of couples formed with POLines. In the relation R, we notice that POLines appears only in pair (POLines, Items). Then POLines forms with the other elements pairs with similarity degrees equal to ϵ (very weak). So, items will have as preference degree:

$max\ [min(0{,}6; 0{,}57), min\ (0{,}6; \epsilon)] = 0{,}57.$

Preference degrees of the other elements will be computed in the same way and we will have:

{Items(0,57), Item(0,6), ItemNumber(0,57), Quantity(0,6), Price(0,6), InvoiceTo(0,62), Adress(0,51), Street(0,7), City(0,7)}.

Thus, from the user preferences on a source schema, we could reduce the research space of mappings. Which will serve to optimize the process of schema matching. We propagated weights assigned to elements of the source schema to discover user preferences over the target schema.
The output scores give an automatic order of the target schema elements based on the interest expressed by the user.

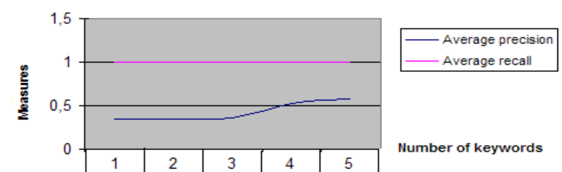
V. Experiments and evaluations

In order to evaluate our approach, we used measures of "Precision" and "Recall":

$$\text{Precision} = \frac{CDM}{DMS}\,; \quad \text{Recall} = \frac{CDM}{CEM}.$$

Where:

CDM represents the relevant determined matches; DMS represents discovered matches by the system; CEM represents all correct existing matches (in our case, matches given by COMA++).
All tests were performed on a corpus of schemas taken from the OASIS[2] web site (http://www.openapplications.org). OASIS is an international consortium whose goal is to promote the adoption of product-independent standards for information formats such as SGML[3], XML, and HTML[4]. This collection is composed of 1000 XSD[5] schemas, with about 220 nodes as number of elements in each schema.
In this work, we supposed that COMA++ provides all existing correspondences. On this basis, we are going to present the different tests achieved.
From a source XSD schema and a query including the keywords and preference degrees of the user, we extracted the FSS from the source schema. The objective here is to provide the user important mappings between preference classes of a source schema and a target ones. For that, we had varied the number of keywords to see the impact of this variation on measures of precision and recall (figure 5).
We suppose that the user have to propose from 1 to 5 keywords (n = 1 ...5). The goal then, is to compute the precision, the recall as well as the execution time while using n number of keywords.
We performed experiments with different combinations of keywords. Thus, for every n, we produced 10 possible combinations. This, allows us to compute the average precision, the average recall and the average execution time for each n.
The results of the experiments are presented in figures 5 and 6.



	1	2	3	4	5
Average precision	0,34	0,34	0,36	0,52	0,58
Average recall	1	1	1	1	1

FIG. 5 –. *Impact of the Variation of keywords number on the average precision and recall.*

The figure 5 shows that if the user chooses a weak number of keywords (less than 3), the average precision is mediocre (between 0.34 and 0.36). From a number of keywords greater than 4, the precision is correct (greater than 0.50).
However, while increasing the number of keywords, the execution time grows logically (see figure 6).
This permits us therefore to conclude that it's necessary to determine a compromise between the number of keywords proposed by the user and the schema matching time.
Experiments on a large scale should be done to determine such compromise.

VI. Conclusion and future work

In this paper, we have proposed a new approach that consists in applying fuzzy sets theory on schemas in order to express

2. Organization for the Advancement of Structured Information Standards

3. Standard Generalized Markup Language
4. Hypertext Markup Language
5. XML Schema Description

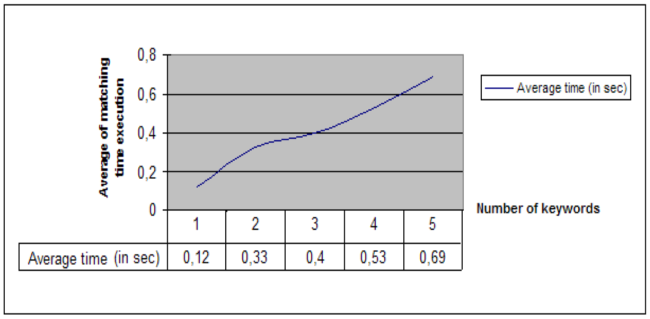


FIG. 6 –. *Impact of the Variation of keywords number on the matching execution time.*

user preferences.

We introduced, as a first main contribution of this paper, the notion of fuzzy subset schema (FSS), that may be over a part of a schema and the notion of generalization of the FSS, that is explicitly defined over the whole schema, using the links between elements of the schema. This will reduce the research space of the schema matching operation.

The method that we have proposed aims to propagate weights to elements of a target schema according to the user preferences over a source one and mappings found by the matcher between the two schemas. Fuzzy relations are used as a basis to define the propagated preferences degrees to the target schema from a source one, which is the second main contribution of this paper. Our approach is a first stage to optimize the process of schema matching.

According to the experiment results, our approach has shown good results but it presents some limits. Indeed, it doesn't allow matching between more than two FSS, it is due to the utilization of COMA++ that performs only two schemas at the same time. Besides, the assignment of preference requires a knowledge of the schema. It is not obvious with the real databases that have complex schemas which are difficult to understand. A solution to this problem consists to work on the schema summary [7]. This summary provides an idea all over the schema.

In future work, we plan to extend this work to define some " fuzzy " views and to construct the mediated schema according to these views. We aim specifying a "fuzzy mediated schema" as a set of "fuzzy views".

Other perspective that deserves to be studied is to extend the notion of the FSS to determine matches dynamically. Correspondences are until now discovered statically, it means that they are found previously before the user query. We propose to study the problem of determining mappings dynamically as the user query arises.

REFERENCES

[1] Aumüller, D., H. Do, S. Massmann, E. Rahm. Schema and Ontology Matching with COMA++. *Proc. SIGMOD (Software Demonstration), Baltimore, 2005.*

[2] Baziz, M. Indexation conceptuelle guidé par ontologie pour la recherche d'information. *PhD Thesis, Institut de Recherche en Informatique de Toulouse, 2005.*

[3] Bernstein, P.A. Generic Model Management: A Database Infrastructure for Schema Manipulation. *Microsoft Research, Microsoft Corporation, 2003.*

[4] Boughanem, M. Contribution à la Formalisation et à la Spécification des Systèmes de Recherche et de Filtrage d'Information. *Habilitation à Diriger les Recherches, Université Paul Sabatier de Toulouse, 2000.*

[5] Boughanem, M. Les Systèmes de Recherche d'Information: d'un modèle classique à un modèle connexionniste. *Thèse de Doctorat de l'Université Paul Sabatier, Toulouse (France), 1992.*

[6] Buche, P., O. Haemmerlé, and R. Thomopoulos. Integration of heterogeneous, imprecise and incomplete data: an application to the microbiological risk assessment. *In Proceedings of the 14th International Symposium on Methodologies for Intelligent Systems, ISMIS'2003, Lecture Notes in Artificial Intelligence, volume 2871, pp. 98-107, Maebashi, Japan, 2003.*

[7] Cong, Y. and H.V. Jagadish. Schema summarization. *In VLDB'06, Seoul, 2006.*

[8] Doan, A., P. Domingos, and A. Halevy. Reconciling Schemas of Disparate Data Sources: A Machine-Learning Approach. *ACM SIGMOD, Santa Barbara, California, USA, pp. 1-12, May 2001.*

[9] Doan, A. , J. Madhavan, P. Domingos, and A. Halevy, Learning to Map between Ontologies on the Semantic Web. *WWW2002, Honolulu, Hawaii, USA, pp. 1-12, May 2002.*

[10] Do, H. and E. Rahm. Coma- A system for flexible combination of schema matching approaches-. *In Proceedings of the 28th Conf on Very Large Databases, 2002.*

[11] Klinker, G., C. Bhola, G. Dallemagne, D. Marques, and J .McDermott. Usable and reusable programming constructs. *Knowledge Acquisition, pp. 117-135, 1991.*

[12] Li, W., C. Clifton. SEMINT: A tool for identifying attribute correspondences in heterogeneous databases using neural networks. *Data & Knowledge Engineering, 33(1), pp. 49-84, 2000.*

[13] Lipo, W., J. Yaochu. Fuzzy systems and knowledge discovery. *Second International Conference, FSKD 2005, china , 2005*

[14] Lu, J., S. Wang and J. Wang. An Experiment on the matching and Reuse of XML Schemas. *In Proceedings of ICWE 2005*.

[15] Madhavan, J., P.A. Bernstein and E.Rahm. Generic schema matching with Cupid. *Proceedings of the 27th VLDB Conference, Roma, Italy, pp. 1-10, 2001.*

[16] Melnik, S., H. Molina-Garcia and E. Rahm. Similarity flooding -a versatile graph matching algorithm-. *In Proceedings of the International Conference on Data Engineering, 2002.*

[17] Miller, R.J., M.A. Hernbndez, L.M. Haas, L. Yan, C.T. Howard Ho, R. Fagin, L. Popa. The Clio Project: Managing Heterogeneity. *SIGMOD Record 30, 1, pp. 7883, March 2001*

[18] Molina-Garcia, H., Y. Papakonstantinou, D. Quass, A.Rajaraman, Y.Sagiv, J.Ullman and J.Widom. The TSIMMIS project-Integration of heterogeneous information sources-. *Journal of Intelligent Inf.Systems 8(2), 1997.*

[19] Nakoula, Y. Apprentissage des modèles linguistiques flous, par jeu de règles ponderées. *PhD Thesis, Université de Savoie., 1997*

[20] Rahm, E. and P.A. Bernstein. A survey of approaches to automatic schema matching,*VLDB journal. Vol.10, pp. 4, 2001.*

[21] Rahm, E. and P.A. Bernstein. On Matching Schemas Automatically. *Microsoft Research Technical Report MSR-TR, pp. 17, 2001.*

[22] Sanchez, E. Fuzzy Logic And the Semantic Web. *479 pages, 2006*

[23] Smithson, M.J. and J. Verkuilen. Fuzzy Set Theory- applications in the social sciences - *97 Pages, 2006*

[24] Tranier, J. , R. Barar, Z. Bellahsene and M. Teisseire. Where's Charlie: Family-Based Heuristics for Peer-to-Peer Schema Integration. *Proc of IDEAS, pp. 227-235, 2004*

[25] Zadeh, L.A. Soft computing and Fuzzy logic. *IEEE Software, pp. 48-56, 1994*

[26] Zadeh, L.A. "Outline of a new approach to Analysis of Complex Systems and Decision Process". *IEEE Trans on Systems, Man, and Cybernetics, pp. 28-44, 1973.*

[27] Zadeh, L.A. , Toward a perception-based theory of probabilistic reasoningvwith imprecise probabilities, *Journal of Statistical Planning and Inference, 105, pp. 233 - 264, 2002.*