Benchmarking XML-Schema Matching Algorithms for Improving Automated Tuning

Mohamed Boukhebouze, Rami Rifaieh**, Nabila Benharkat*, Youssef Amghar**
** LIRIS, National Institute of Applied Sciences of Lyon, Lyon, France*
***San Diego Supercomputer Center, University of California San Diego, USA*
{mohamed.boukhebouze, nabila.benharkat, youssef.amghar}@insa-lyon.fr
rrifaieh@sdsc.edu

Abstract

Several matching algorithms were recently developed in order to automate or semi-automate the process of correspondences discovery between XML schemas. These algorithms use a wide range of approaches and matching techniques covering linguistic similarity, structural similarity, constraints, etc. The final matching combines arithmetically different results stemmed from these techniques. The aggregation of the results uses often many parameters and weights to be adjusted manually. Generally, this task is achieved by human experts and requires a perfect understanding of the matching algorithm. In order to reduce the human intervention and improve matching quality, we suggest automating the tuning of the various structural parameters used within XML-Schema matching algorithms. In this work, we offer a benchmark, for three tools, that seeks mathematical relations between parameters values and schema topology. In consequent, we propose an algorithm for the tuning of these parameters for studied tools.

Keywords: Matching, XML Schemas, Benchmark, Automatic Tuning.

1. Introduction

Information systems are evolving toward dynamic environment where requirements and data are changing constantly. Different industries are bringing their system toward openness and adopting flexible strategies to cope with their new challenges. P2P and loosely coupled architecture are gaining more space versus warehousing and traditional integration in the new market terminology. Obviously, these changes entail a constant evolution at the schema level. Therefore, mapping techniques become crucial tools to keep up with the constant change of data internal format and its delivery.

Even though, schema mapping is very attractive area of research, very little was developed as real scale solutions. Projects such as Microsoft Protoplasm [20], and IBM Garlic [21] have attracted much attention even with limited capability. In addition, many challenges, in automating schema matching and its related issues, are not completely covered.

For instance, several algorithms were proposed for automating the process of matching, (e.g. COMA [1], EXSMAL [6] and SCIA [7]). These algorithms combine various techniques (structural, linguistic, etc…). Each technique has a numeric weight also called parameter. The final result combines the various results from these techniques. The choice of these parameters is important to reach a good precision with the matching task. However, the adjustment of these parameters is done manually by the developers of these algorithms. In addition, no empirical study has covered multiple algorithms with an extended number of mapping schemas. Moving toward a full-scale solution for schema matching/mapping has to deal with these preceding challenges.

In our work we are trying to automate the tuning of the structural parameters used within the matching systems. This will help us to reduce consequently the human intervention and also to reach a good matching quality (i.e. precision, recall, overall). The approach suggested in this paper covers the benchmarking of matching algorithm and the tuning of structural parameters. We propose an algorithm that uses mathematical relations discovered with our empirical study (benchmark) to tune the matching tools. The work is carried for XML Schemas and is based on discovering linear relations between entered schema's topology and the values of structural parameters.

This paper is organized as follows. We start the section 2 with a state of the art covering matching algorithms, existing benchmarks, and tuning techniques. We present, in section 3, our approach of benchmarking and tuning. In section 4, we describe the phases of the benchmark and its application to selected matching algorithms. In section 5, we outline the automatic tuning algorithm along with the results of using it. We wrap the paper with a brief conclusion and some directions of future works.

2. State of the art

To reach our goals, we have divided our interest into three categories: how to select matching algorithm, how to benchmark matching algorithm, and finally how to tune matching algorithm.

The first step in our work consists of collecting a set of matching algorithms. In the literature, there are several types of these algorithms. Some algorithms work with the schema such as Cupid [4], Similarity Flooding [5], COMA [1], EXSMAL [6] and SCIA [7]. Others work with the data instance such as XMapper [10] and CLIO [11] or with ontologies such as FALCON [21] and QOM [12].

There exist several classifications [1], [2], [8], [9] for comparing these algorithms. Most complete and the most quoted is the one proposed in [2]. A revision of this classification is proposed in [1] where the authors added more discriminatory criteria. The most recent classification is proposed in [9] where authors consider only the approaches based on diagrams.

In our work, we are particularly interested with algorithms treating XML schema. Instead of blindly picking a set of matching algorithms that we will work with. We use the classifications in [9] to identify this set based on similar characteristics. In this respect, SCIA [7], EXSMAL [6] and COMA [1] work very similarly and use comparable matching techniques based on: string matching, linguistic resources, constraints, and graph structure.

Secondly, the evaluations of the matching algorithms are made individually. In order to propose a benchmark, we have initially studied the individual evaluations of the matching systems provided for COMA [1], SF [5] and the evaluations done in [13]. The latter compares tools based on 5 criteria (Input, Output, quality, effort, and time). The first 4 criteria was previously identified in [14]. The quality (i.e. Precision and Recall) and effort (i.e. Overall) are measured based on a referential of manual matching.

The tools examined in [13] are COMA, Cupid, and Similarity Flooding (SF) using real defined schemas. It uses six XML schemas and two SQL DDL schemas with element number varying between 10 and 80. This benchmark is not sufficient because it doesn't take into account all entry schema characteristics (number of the paths and depth).

The evaluation of SF [5] seeks the relations between various filters and parameters of the algorithms influence on the matching results. It includes 9 matching tasks definite from eighteen schemas (XML and SQL DDL) with a number of elements between 5 and 22. The evaluation of COMA [1], uses 5 XML schemas with various depth, number paths, and number of nodes (ranging from 40 to 140).

None of the preceding works can be really called a full-scale benchmark. We consider very interesting, for our work, to compile a benchmark of matching for the tools previously selected. This benchmark should compare the effectiveness of the algorithms doing the same task. Consequently, it will help us to deduce a set of rules for the automation and the optimization of the parameters during the tuning task.

Thirdly, in the majority of the schemas matching systems, the users carry manually the task of tuning. The tuning consists in adjusting the parameters of different matchers. The difficulty of this task is due to the number of parameters to be adjusted, and often requires important expertise and full understanding of the matching algorithm. In the literature, very little effort was given to this question beside the work in [15] and [16].

In [15], the authors focused on the linear problem of combining matching results. In order to evaluate the different matchers combination, the authors checked the correlation between the weights suggested by genetic algorithms and the precision of the mapping. The use of genetic algorithms makes it possible to have a generic solution for the problem of tuning and resolving the linear combination of matching results (i.e. finding the weight for each matcher). However the success of genetic algorithms to find an optimal solution highly depends on a physical function that uses a combination of predefined functions. This solution can't be applied to our approach since we are aiming to discover these functions between the structural parameters.

The eTuner [16] is an approach of automatic tuning for matching systems. Its principal idea is to synthesize a collection of matching scenarios implying a schema S, for which we know already the matching, and then employ this collection to parameterize the system M. The eTuner offers a good solution to the problem of the automatic tuning. However, it passes by two steps including the disturbance of the schemas and execution of the tuning to carry out the parameters. This requires creating a synthesis scenario for each schema and which consumes a considerable cost in time and effort.

3. Suggested Approach

With the vulgarisation of XML syntax, many applications have become dependent on XML and its Schema representation. In many cases, developers need to specify a set of similarities and correspondences between independent applications using XML Schema. Schema matching is a very promising technique developed to simplify the developers' task of creating a mapping between schemas/representations.

Many approaches, tools and algorithms were suggested to tackle the process of schema mapping.

This includes various algorithms for schema matching and correspondences discovery. However, existing algorithms share a common weakness in term of needed expertise to run and verify the results of matching. Indeed, a matching algorithm is written by a developer with marginal knowledge in the field of application. On one hand, the developer understands all the trumps of the algorithm and knows exactly how to optimize it for a specific task. On the other hand, the developer has minimal contribution in the post-matching task where domain expert excel. In addition, user (or domain expert) can barely understand the matching algorithm and its parameters. Left alone with standard parameters, the user can't reach better quality results for their matching task.

In our work we are trying to automate the tuning of parameters existing within the matching systems. This will help in reducing the human intervention, from developers and users, and also to reach a good matching quality (precision, recall, overall). ¶Indeed, our approach consists in finding the best parameters to use according to the studied schemas and their characteristics.

In particular, we believe that schema topology (number of nodes, number of paths, depth, etc.) is directly related to the quality of the matching with respect to structural parameters. Indeed, matching algorithm uses structural parameters (e.g. P_Parents, P_Children, P_Leaves,etc. for COMA) that can influence the matching results. In order to discover some of the rules that exist between the schema topology and the values of these parameters, we propose the following approach.

Firstly, we start by applying a series of experiments, hereafter called benchmark, with three selected algorithms (COMA [1], EXSMAL [6], SCIA [7]). We collect schemas and create manual correspondences between them. Afterward, we repeatedly apply the automatic matching on the schemas in the aim of finding a set of consistent mathematical model (or relations). The consistency of the relations with a topological pattern can help us refine the values of structural parameters.

Secondly, we create a data store that includes all the rules empirically deduced form the benchmark. We define after a tuning algorithm that uses these stored relations and apply them on new schemas. In other words, the algorithm will seek in the entered schemas some topological patterns similar to those stored in the data store and suggest the best set of parameters' values for a specific algorithm.

The two steps of our approach are depicted in Figure.1. The input of the first step is a set of XML Schema and a set of tools. At first, the benchmark identifies the structural parameters to study with each tool. Then it tries to apply a common methodology using heavily data analysis and statistical methods to verify and test models. The output of the first step is a set of acceptable models (also called rules or relations throughout the paper). The second part of our approach is interested in suggesting values to be used for a specific matching task. The Input of this step is the set of models (stored in the data store), the matching schemas, and the tool. The Output is the suggested value for each structural parameter involved in the tool.

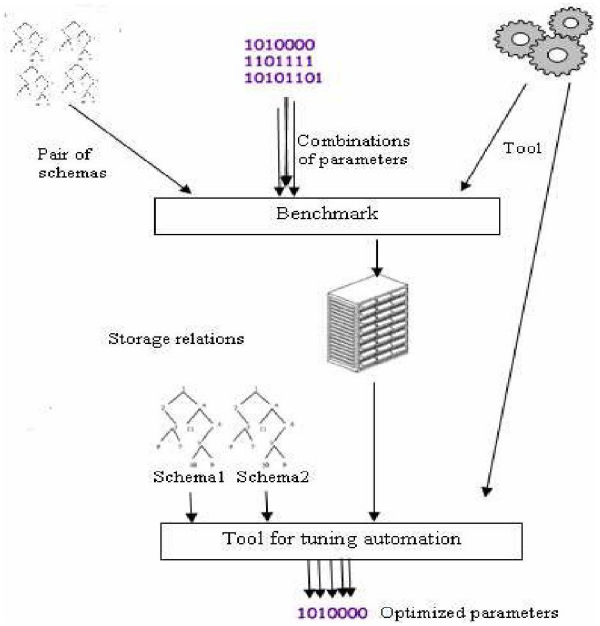

Figure.1. Steps of the Approach

4. Benchmark

The goal of this benchmark is to deduce a set of rules between schema topology (e.g. number of element, depth, number of path, etc.) and the parameters of structural similarity within various matching algorithms. The benchmark covers 3 tools SCIA [7], EXSMAL [6] and COMA [1] previously described in the related works. As mentioned, the choice of these tools is based on similar characteristics as identified in [9].

4.1 Common Methodology

For each selected tool and each pair of schemas, we start with identifying the possible set of structural parameters by incrementing each one of them with a certain delta. We calculate, then, three quality values: Precision, Recall and Overall. The following step consists in keeping only the combinations of parameters which give a better Precision, Recall, and Overall. These refined parameters will be used after within a data analysis method. This method will enable us to find a linear relation between the values for the tested matcher. At the end of this step, we

validate the found relation by applying methods of test significance such as the assumption test and the test of Fisher. If the tests are valid, we call the linear relation between parameters a model and we store it in a data store. In the contrary, we try to apply other methods of analysis of data. Figure 2 schematizes the steps of our benchmark.

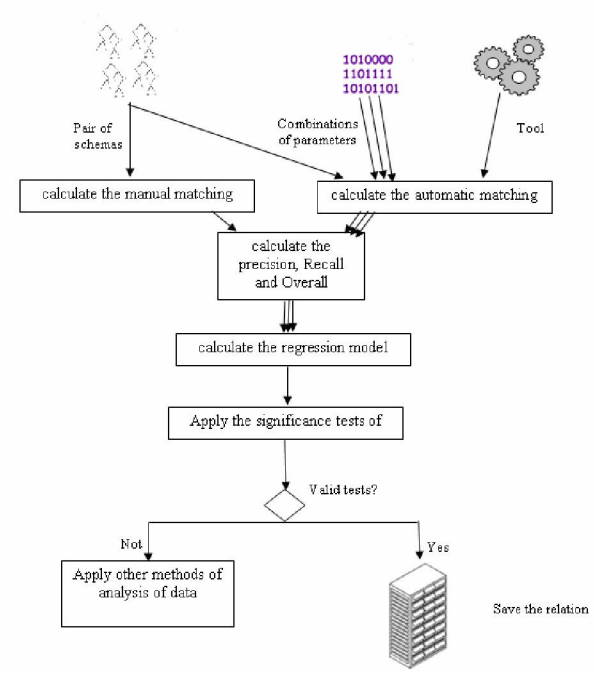

Figure 2. Steps of the Benchmark

4.1.1 Choosing Schemas:

The majority of schema matching systems, including our selected tools, can be used with XML Schema. Therefore, we chose to restrict our benchmark by using only a set XML Schema. The selected schemas are extracted from the web and reflect real use in the *Bibliographical* filed. We chose (n=20) schemas with a number of elements and a number of paths included between [1,80], and having a depth ranging between [2,7]. We then created 190 pairs of schema from all possible combination $C_n^k = \frac{n!}{k!(n-k)!}$ for n=20 and k=2.

4.1.2 Selecting Parameters

The XML Schemas are characterized by some topological facet such as the number of paths, number of elements, and maximum depth. These three topological facets are used in our benchmark. In addition, the tools use a set of parameters that we are seeking their correlation with respect to schema characteristics.

In order to automate the structural coefficients for the selected tools, we will search the best set of coeff_ancestor, coeff_cibling, coeff_immediate_disendante, coeff_leaf) for EXSMAL, the best set of (P_Parents, P_Children, P_Leaves, and P_ Siblings) for COMA, and the best set of PATH_WEIGHT_COMB, GRAPH_WEIGHT_COMB , and STRUCT_WEIGHT_UPDATE_ANCESTOR for SCIA.

4.1.3 Measuring Qualities

In order to compare the quality of the matching, we have established a manual matching as referential. Therefore, the results obtained by the automatic matching are separately checked with respect to three quality measurements: Precision, Recall and Overall.

The Precision and Recall are largely used in the field of the information retrieval and they are also used in the evaluations of matching systems in [3]. The Overall is developed specifically in the context of schema matching. It measures the effort of post-matching necessary to add the true negative and remove the false positive. The following formula are used to calculate these measurements:

Precision = $\frac{|B|}{|B|+|C|}$, calculates the number of true correspondences $|B|$ found among those returned ($|B|+|C|$); $|C|$ is called false positive.

Recall = $\frac{|B|}{|A|+|B|}$, calculates the number of true correspondences $|B|$ found among the total of true correspondences ($|A|+|B|$); $|A|$ is called true negative.

Overall = $1 - \frac{|A|+|C|}{|A|+|B|} = \frac{|B|-|C|}{|A|+|B|} = \mathrm{Re}call*(2 - \frac{1}{precision})$, represents the effort needed to correct the results of an automatic matching (i.e. adding the true negative and removing the false positive).

4.1.4 Analyzing Results

In order to analyze the results obtained from the measurements of automatic matching, we will employ some techniques used for data analysis. However, data analysis covers various methods [18] such as data compression and data approximation. The analysis by data compression makes it possible to replace a voluminous set of data into a reduced one by minimizing the loss of information. This type of analysis (e.g. ACP [19]) is particularly relevant for data streaming, images, audio, and video. The analysis by the data approximation works with a transformed representations of the data [18]. Two techniques are mostly used in this category regression and clustering. The multiple regression and the nonlinear regressions use a model to estimate the data and the histograms. Clustering and the selection by random pulling provide a reduced representation of the data.

For our benchmark, we are seeking the relations between the studied structural parameters and not interested in reducing the number of these variables (parameters). Therefore, the application of linear regression model [19] seems to be the best method for finding a relation between a set of parameters.

$$(E) \quad Y = a_0 + a_{i1}x_{i1} + a_{i2}x_{i2} + \ldots + a_{ij}x_{ij} + \ldots + a_{ip}x_{ip} + e_i \quad i \in \{1 \ldots n\}$$

This equation is a hyperplane equation with p dimensions. The parameters a_0, a_1, $a_2 \ldots a_p$ are called "coefficients of regression ".

The formulas of the multiple linear regression are in the matrix, $Y = XA + e$, as described in [18]:

$$X = \begin{bmatrix} 1x_{11} \ldots x1p \\ 1x_{21} \ldots x_{2p} \\ \ldots \\ 1x_{n1} \ldots x_{np} \end{bmatrix}, \; Y = \begin{bmatrix} Y_1 \\ \cdot \\ Y_n \end{bmatrix}, \; A = \begin{bmatrix} a_0 \\ \ldots \\ a_p \end{bmatrix} \; et \; e = \begin{bmatrix} e_1 \\ \ldots \\ e_n \end{bmatrix}$$

To calculate matrix A of the parameters, we should apply: $A = (X^t X)^{-1} X^t (Y - e)$.

4.2 Application of the benchmark for EXSMAL:

The execution of the benchmark on EXSMAL seeks to determinate the best values of parameters <coeff_anc, coeff_fr, coeff_fimm, coeff_feuille > with respect to the entry schemas characteristics (number of elements, number of paths, max depth). In EXSMAL the following formula should always hold:

(i) $coeff\_anc, coeff\_fr, coeff\_fimm, coeff\_feuille \in [0,1]$
(ii) $coeff\_anc + coeff\_fr + coeff\_fimm + coeff\_feuill\ e = 1$

Therefore, we have calculated all the combinations of these four parameters by increasing each one of them with a chosen delta, =20%. We have obtained 20 possible combinations after rejecting combinations about of boundary (i) and violating sum (ii). After that we have carried out the matching operation and calculated the Precision, Recall and Overall. This operation was done for each of the 20 combinations with the 190 pairs of chosen schemas. We kept only the combinations for which we have the best quality. Figure.4 illustrates the results found for a pair of schemas having a number of elements equal to 8 and 12, a number of paths equal to 4 and 7 and depth of 3 and 4. The rows in yellow color represent the combinations that we will keep for the data analysis step.

For data analysis, we have applied the multiple regression to find a linear relation between the four structural parameters. By calculating the coefficients of regression, we suppose that the parameters are linearly dependent. This linear relation is later tested to verity the assumption. The calculation of the regression coefficients is resumed as follows:

We firstly consider coeff_anc as explained variable and coeff_fr, coeff_fimm as explanatory variables, therefore the linear regression equation becomes:

$$coeff\_anc = a_0 + a_1\ coeff\_fr + a_2 Coeff\_fimm$$

The *coeff_feuille* will be calculated by using (ii):
1 - coeff_anc + coeff_fr + coeff_fimm.

The value of a_0, a_1 and a_2 are calculated using the matrix A. We found $a_1 = 0{,}35$ $a_2 = 0{,}35$ *and* $a_0 = 0$ thus :
(R1) coeff_anc= 0,3 5Coeff_fr + 0,35 Coeff_fimm

Coeff_anc	Coeff_fr	Coeff_fimm	Coeff_feuille	Precision	Recall	Overall
0,1	0,1	0,1	0,7	0,23	1	-2,28
0,1	0,1	0,3	0,5	0,23	1	-2,28
0,1	0,1	0,5	0,3	0,23	1	-2,28
0,1	0,1	0,7	0,1	0,23	1	-2,28
0,1	0,3	0,1	0,5	0,23	1	-2,28
0,1	0,3	0,3	0,3	0,23	1	-2,28
0,1	0,3	0,5	0,1	0,23	1	-2,28
0,1	0,5	0,1	0,3	0,23	1	-2,28
0,1	0,5	0,3	0,1	0,23	1	-2,28
0,1	0,7	0,1	0,1	0,18	1	-3,55
0,3	0,1	0,1	0,5	0,23	1	-2,28
0,3	0,1	0,3	0,3	0,23	1	-2,28
0,3	0,1	0,5	0,1	0,23	1	-2,28
0,3	0,3	0,1	0,3	0,18	1	-3,55
0,3	0,3	0,3	0,1	0,18	1	-3,55
0,3	0,5	0,1	0,1	0,18	1	-3,55
0,5	0,1	0,1	0,3	0,18	1	-3,55
0,5	0,1	0,3	0,1	0,18	1	-3,55
0,5	0,3	0,1	0,1	0,18	1	-3,55
0,7	0,1	0,1	0,1	0,18	1	-3,55

Figure 4. Possible combination for one pair of schemas and the quality results of their matching.

To validate the found relation, we calculated the standardized residues. The results are presented in Figure 5. We note that all the values of the standardized residue ranging between [-3,3] what shows that the errors are acceptable [19]. To appreciate the contribution of the explanatory variable (i.e. coeff_fr, coeff_fimm in (R1)) we have calculated the coefficient of determination $R^2 = \sum_{i=1}^{n} (\hat{Y}_i - \overline{Y})^2 \Big/ \sum_{i=1}^{n} (Y_i - \overline{Y})^2$. This helps to explain the fluctuations of the dependent variables, in the regression equation. We have found the value of $R^2 = 0{,}79$ (relatively high). This means the adjustment of the regression straight line at the experimental points is of good quality.

Finally, to confirm the assumption of linear dependency between the parameters, we have applied the Fisher test as following:

(1) Statistical Assumption:
H0: the coeff_anc is linearly independent of the coeff_fr and coeff_fimm.
(2) Threshold of significance: represents the risk of rejecting wrongly a true assumption H 0. We have chose the threshold of significance = 0,05.
(3) Conditions & Population: our condition, for the test, is that random samples coming from a normal distribution does not let suspect of nonlinear relation between parameters (coeff_anc, coeff_fr and coeff_fimm).
(4) Choosing statistical method: we have chose to apply F with the preceding condition. $F=R^2(n\ \ p\ \ 1)/(1\ \ R^{2\,2})p$ where p is the number of explanatory variables+1 and n is the number of observations.
(5) Rule of decision: the critical value of F is F_{RMc}=0,03. So we have adopted to reject H0 if F>0,03.
(6) Value of reduced variation: from the results of the sample by replacing suitable values in the relation F, we obtain the value following for F = 5,67
(7) Decision and conclusion: it compares the numerical value obtained for the variation reduced with the rule of decision adopted into 5. The value F= 5,67, H 0 is rejected. So there is a significant linear correlation between the coefficients coeff_anc, coeff_fr and coeff_fimm

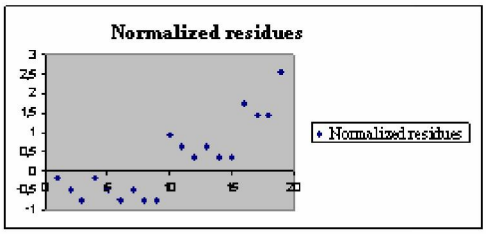

Figure 5. Values of the moralized residues calculated for the relation (R1)

After having confirmed the validity and the test significance, we have accepted the relation (R1).

As a remainder, this relation is found for one pair of schemas having a number of elements equal to 8 and 12, the number of paths equal to 4 and 7 and depths 3 and 4. In order to find another possible relations with other pairs of schemas, we have repeated the preceding work 190 times (number of possible schema pair from the 20 selected schemas). This procedure has generated 5 relations (i.e. models) as illustrated in Figure 6.

Relation	Mathematic relation	N and P	D	test numbers
R1	Coeff_anc = 0.35Coeff_fr + 0.35Coeff_fimm	]0,80]	[2,7]	173
R2	Coeff_anc = 0.33Coeff_fr + 0.33Coeff_fimm	]0,80]	[2,7]	9
R3	Coeff_anc = 0.25Coeff_fr + 0.04Coeff_fimm	]0,80]	[2,7]	4
R4	Coeff_anc = 0.13Coeff_fr + 0.27Coeff_fimm	]0,80]	[2,7]	3
R5	Coeff_anc =2.2Coeff_fr + 0.45Coeff_fimm	]0,80]	[2,7]	1

Figure 6. Found EXSMAL models for the 190 pairs of selected schemas

As illustrated in Figure 7, the relation (R1) has repeatedly appeared with 91 % of the time, of the total for 190 pairs of tested schemas. Other relations have very low rate of acceptance in large scale (the closest is (R2) with 4.74% of rate).

We have concluded that the relation (R1) is the most significant. We have saved (R1) in our models datastore and it will be used for the automatic tuning. However, the applicability of (R1) is still limited to schemas having the number of elements and paths varying between [1,80] and a depth varying between [2,7].

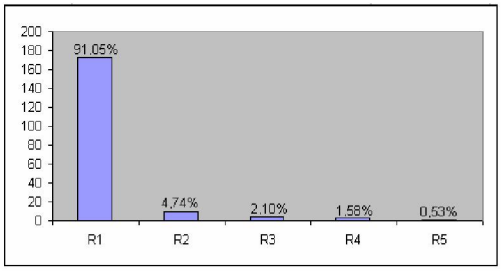

Figure 7. Percentage of validation of found models when applied to the 190 pairs of selected schemas

4.3 Application of the benchmark for COMA:

In order to discover the set of relations between the structural parameters used in COMA (P_Parents, P_Children, P_Leaves, and P_ Siblings), we have applied the same methodology described for EXSMAL.

The acceptable values for each structural parameter, in COMA, are included in the interval [1,10]. We have calculated the combinations of the four parameters by increasing each with =20%. We have found 625 possible combinations. This number is considerably higher than 20 combinations, found in EXSMAL, since no equation similar to (ii) exists for COMA.

We have repeated the same work done for EXSMAL. We have calculated the Precision, Recall and Overall for each of the 625 combinations with the 190 pairs of selected schemas. Finally, we kept only the combinations with best quality for calculating the regressions models.

We have found four relations or models as depicted in Figure 8. After validating these four relations with the tests of significance and Fisher, the relation (R'1) was saved to be used by the automatic tuning algorithm.

Relation	Mathematic relation	N and P	D	test numbers
R1	P_Parents = 0,6 P_Children + 0,6 P_Leaves	]0,40[	[2,4]	94
R2	P_Parents = 0,48 P_Children + 0,48 P_Leaves	[40,80]	[2,4]	27
R3	P_Parents = 0,11P_Children + 0,11 P_Leaves	]0,40[	[5,7]	51
R4	P_Parents = 0,07 P_Children + 0,07 P_Leaves	[40,80]	[5,7]	18

Figure 8. Found models of COMA for the 190 pairs of selected diagrams

4.4 Application of the benchmark for SCIA:

In order to discover the set of relations between the structural parameters used in SCIA (PATH_WEIGHT_COMB, GRAPH_WEIGHT_COMB and STRUCT_WEIGHT_UPDATE_ANCESTOR), we have followed the identical methodology described for EXSMAL and COMA.

We have calculated the combinations of the four parameters by increasing each one of a delta of 20%. We have found 125 possible combinations. We have carried out the matching and calculated the Precision, Recall and Overall for each of the 125 combination with the 190 pairs of selected schemas. After validating the relations, we have found four models for the structural parameters of SCIA as illustrated in Figure 9.

Relation	Mathematic relation	N and P	D	test numbers
R1	PATH_WEIGHT = -0,23 RAPH_WEIGHT - 0,62 STRUCT_WEIGHT+0,51	]0,40[	[2,4]	94
R2	PATH_WEIGHT = -0,5 RAPH_WEIGHT - 0,5 STRUCT_WEIGHT+0,5	[40,80]	[2,4]	27
R3	PATH_WEIGHT = -0,25 RAPH_WEIGHT - 0,75 STRUCT_WEIGHT+0,55	]0,40[	[5,7]	51
R4	PATH_WEIGHT = -0,49 RAPH_WEIGHT - 0,33 STRUCT_WEIGHT+0,52	[40,80]	[5,7]	18

Figure 9. The SCIA models found for the 190 pairs of selected schemas

After applying the benchmark and finding the grail for the studied algorithms, we will show in the next section how to use the models.

5. The Algorithm of Automatic Tuning:

We are interested in automating the task of adjustment for the various structural parameters used within the studied matching systems. For that we proposed an algorithm for the tuning automation that uses the results of benchmark. Indeed, this algorithm not only reduces the user intervention but also offers a better precision for the matching task.

The algorithm of automatic tuning has as input two schemas and the tool that we want to optimize its structural parameters. The algorithm initially will calculate the number of elements, numbers paths and the depth of each diagram. Then it will choose, among the relations saved, from the benchmark, the relation to be applied with respect to the characteristics of the schemas. Finally the algorithm calculates and displays the suggested values. Figure 10 illustrates the input and the output of the algorithm of automatic tuning. The algorithm is described in the following pseudo-code:

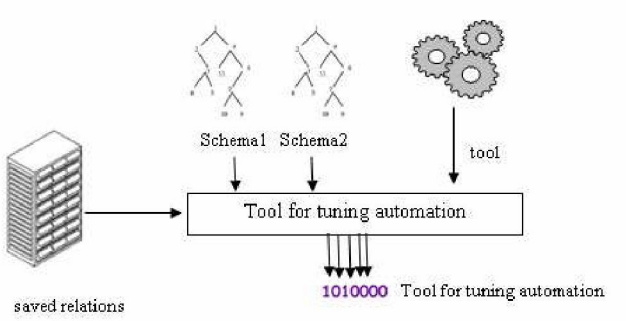

Figure 10. Input and output of the tuning automation tool

```
Tuning_Algorithm (schema1, schema2 ,tool)
Begin
  <N1,P1,D1> = Calculate  the caractéristics(schema1)
  <N2,P2,D2> = Calculate the caractéristics(schema2)
  if (tool = 'EXSMAL' )
       and (N1, N2, P1 and P2    ]0,80] )
        and (D1,D2     [2,7]) then
    // generate a random number    [0,1]
    Coeff_fr = generate_random_number (0,1)
    Coeff_fimm = generate_random_number (0,1)
    (a1,a2)<- Query_the_Relations_DB('EXSMAL',
                N1,N2,P1,P2,D1,D2)
    Coeff_anc= 0,35* Coeff_fr + 0,35 * Coeff_fimm
    Coeff_feuille = 1 - Coeff_anc + Coeff_fr
                    + Coeff_fimm
    Write(Coeff_anc, Coeff_fr, Coeff_fimm,
                Coeff_feuille )
  Else Write("No known characteristics of diagrams")
  End_if
  if (tool = 'COMA')
        and (N1, N2,P1) and (P2      ]0,80])
        and (D1,D2      [2,7]) then
    P_Children = generate_random_number (0,100)
    P_Leaves = generate_random_number (0,100)
    P_ Siblings = 10
    (a1,a2)<- Query_ the_Relations_DB ('COMA',
                N1,N2,P1,P2,D1,D2)
    //for N1 and P1      ]0,40[ and D1      [2,4] we have
    P_ Parents = 0,6 P_Children + 0,6 P_Leaves
    // for N1 and P1      [40,80] and D1      [2,4] we have
    P_ Parents = 0,48 P_Children + 0,48 P_Leaves
    // for N1 and P1      ]0,40[ and D1      [5,7] we have
    P_ Parents = 0,11P_Children + 0,11 P_Leaves
    // for N1 and P1      [40,80] and D1      [5,7] we have
    P_ Parents = 0,07 P_Children + 0,07 P_Leaves
    Write(P_ Parents, P_Children, P_Leaves, P_ Siblings)
  Else Write ("No known characteristics of diagrams" )
  End_If
  If (tool = 'SCIA')
      and  (N1, N2,P1,P2     ]0,80])
      and  (D1,D2     [2,7]) Then
    GRAPH_WEIGHT = Generate_random_number (0,1)
    STRUCT_WEIGHT = Generate_random_number(0,1)
    (a1,a2)<- Query_ the_Relations_DB ('SCIA',
                N1,N2,P1,P2,D1,D2)
```

```
    // for N and P    ]0,40[ and D    [2,4] we have
    PATH_WEIGHT = -0,23 RAPH_WEIGHT
                - 0,62STRUCT_WEIGHT+0,51
    // for N and P    [40,80] and D    [2,4] we have
    PATH_WEIGHT = -0,5 RAPH_WEIGHT
                - 0,5 STRUCT_WEIGHT+0,5
    // for N and P    ]0,40[ and D    [5,7] we have
    PATH_WEIGHT = -0,25 RAPH_WEIGHT
                - 0,75 STRUCT_WEIGHT+0,55
    // for N and P    [40,80] and D    [5,7] then
    PATH_WEIGHT = -0,49 RAPH_WEIGHT
                - 0,33 STRUCT_WEIGHT+0,52
    Write(PATH_WEIGHT, GRAPH_WEIGHT
                ,STRUCT_WEIGHT)
  Else Write ("No known characteristics of diagrams" )
  End_if
End
```

5.1 Final Results

We applied a final test with COMA to check the results coming from the tuning algorithm. The experiment consists in applying COMA to match between two given schemas (never used in the test before). We calculated the precision and Recall of the results obtained, figure 11 shows that the precision is equal to 0,23 and that Recall is equal to 0,6 using standard value for structural parameters. We have repeated the same matching after using the value of parameters suggested by the tuning algorithm. The precision has improved to 0.33 and Recall improved to 0,8 ; matching shown in figure 12.

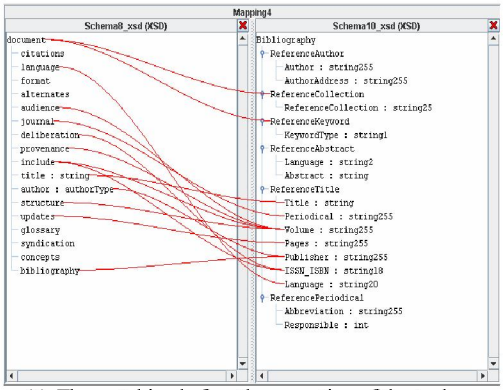

Figure11. The matching before the execution of the tool

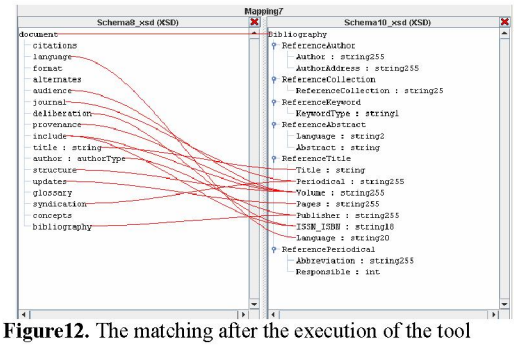

Figure12. The matching after the execution of the tool

6. Conclusion and perspectives:

In order to automate the tuning of various structural parameters of matching systems, we proposed an approach consisting of two stages:

The first step includes carrying out a benchmark to empirically deduce the relation existing between the characteristics of the entry schemas and the structural parameters of the studied tools. Our benchmark covers 20 schemas and a combination set of 190 pairs. We identified three matchers pertaining to the same category: COMA, EXSMAL and SCIA. After carrying an automatic matching for each combination of the parameters values, we keep only the combinations of parameters with high quality (Precision, Recall, and Overall). To validate, at the end of this phase, the found relation we carry out a series of tests of significance. If the tests are valid we save the models found in a database. In the contrary case, we search to apply other methods of analysis of data.

The second step of our approach consists in automating the task of adjustment of the various structural parameters of the studied matching systems. For that we proposed and implemented a tuning automation algorithm that uses the results of the benchmark seen in the preceding section. We have done an experiment to evaluate the precision of our algorithm. This experiment showed that the optimal values suggested by the algorithm lead to a better precision and better Recall.

Our proposal contributes in the solution of the automatic problem of the structural parameters tuning for the matching algorithms. We estimate that a robust solution to this problem should take in consideration, supplement, the tuning of the linguistic parameters and take into account also the tuning of matching algorithm applied to the relational schema.

7. References

[1] Hong Hai Do, Erhard Rahm: COMA - A System for Flexible Combination of Schema Matching Approaches. VLDB 2002: 610-621

[2] Erhard Rahm, Philip A. Bernstein: A survey of approaches to automatic schema matching. VLDB J. 10(4): 334-350 (2001)

[3] David Aumueller, Hong Hai Do, Sabine Massmann, Erhard Rahm: Schema and ontology matching with COMA++. SIGMOD Conference 2005: 906-908

[4] Jayant Madhavan, Philip A. Bernstein, Erhard Rahm: Generic Schema Matching with Cupid. VLDB 2001: 49-58

[5] Sergey Melnik, Hector Garcia-Molina, Erhard Rahm: Similarity Flooding: A Versatile Graph Matching Algorithm and Its Application to

Schema Matching. ICDE 2002: 117-128
[6] Rami Rifaieh, Uddam Chukmol, Aïcha-Nabila Benharkat: A Matching Algorithm for Electronic Data Interchange. TES 2005: 34-47
[7] Guilian Wang, Young-Kwang Nam, and Kai Lin. Critical Points for Interactive Schema Matching, Technical Report CS2004-0779, UCSD Department of Computer Science, January 2004;
[8] Shvaiko, Pavel. A Classification of Schema-Based Matching Approaches. Technical Report DIT-04-093, Informatica e Telecomunicazioni, University of Trento (2004)
[9] Shvaiko, Pavel and Euzenat, Jerome, A Survey of Schema-based Matching Approaches. Technical Report DIT-04-087, Informatica e Telecomunicazioni, University of Trento.(2005)
[10] Lukasz A. Kurgan, Waldemar Swiercz, Krzysztof J. Cios: Semantic Mapping of XML Tags Using Inductive Machine Learning. ICMLA 2002: 99-109
[11] Miller, R.J. et al. The Clio Project: Managing Heterogeneity. SIGMOD Record 30:1: 78–83, 2001
[12] M. Ehrig and S. Staab. Qom - quick ontology mapping. In Proceedings of the 3rd International Semantic Web Conference (ISWC 2004), LNCS 3298, Hiroshima, Japan, page 683,697.
[13] M.Yatskevich. Preliminary Evaluation Of Schema Matching Systems, University of Trento Technical Report # DIT-03-028, November 2003
[14] Hong Hai Do, Sergey Melnik, Erhard Rahm: Comparison of Schema Matching Evaluations. Web, Web-Services, and Database Systems 2002: 221-237
[15] S.Berkovsky, Y.Eytani, Measuring the Relative Performance of Schema Matchers, Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)
[16] Mayssam Sayyadian, Yoonkyong Lee, AnHai Doan, Arnon Rosenthal: Tuning Schema Matching Software using Synthetic Scenarios. VLDB 2005: 994-1005.
[17] Ningsheng Jian, Wei Hu, Gong Cheng, Yuzhong Qu: FalconAO: Aligning Ontologies with Falcon. Integrating Ontologies 2005
[18] Ian T. Jolliffe, " Principal Component Analysis", Springer (ed), ISBN: 0387954422, 487p
[19] Leona S. Aiken, Stephen G. West, Patricia Cohen, Jacob Cohen "Applied Multiple Regression/Correlation Analysis for the Behavioural Sciences", Lawrence Erlbaum Associates(ed), ISBN: 0805822232, 728p.
[20] Protoplasm : Philip A. Bernstein, Sergey Melnik, Michalis Petropoulos, Christoph Quix: Industrial-Strength Schema Matching. SIGMOD Record, Vol. 33, No. 4, December 2004
[21] M.Carey et al. "Towards Heterogeneous Multimedia Information Systems : The GArlic Approach" Technical Report, IBM Almaden Research, 1995.