Evaluating Scalable Matching Tools: a qualityoriented approach

Claudia Rodriguez, Sana Sellami

Evaluating Scalable Matching Tools: a quality-oriented approach

Claudia C. Gutiérrez Rodríguez
I3S Laboratory
Nice Sophia-Antipolis University
Sophia-Antipolis, France
cgutierr@i3s.unice.fr

Sana Sellami
LSIS Laboratory
Aix-Marseille University
Marseille, France
sana.sellami@lsis.org

Abstract— **Actually, the evaluation of matching tools is an entire, complex and complicated research subject which we are interested in. Complex because matching systems can regroup several matching techniques and complicated considering their multiple users. Considering quality as an important element to define, use and evolve particular systems (as information and manufacturing systems), we extend traditional approaches and we propose an evaluation approach based on software product quality principles. In this paper, we offer an evaluation method based on a quality model (characteristics, sub-characteristics, measures…) adapted to the specificities of scalable matching tools. To illustrate our approach, we provide some evaluation results over two scalable matching tools COMA++ and PLASMA.**

D.2.10.h **[Quality analysis and evaluation];** H.2.1.c **[Database integration]**

I. INTRODUCTION

In the last few years, more and more matching tools have being proposed to identify semantic correspondences between structures or models as XML or database schemas and ontologies (i.e. COMA++ [1], Falcon-Ao[2], Protoplasm [3]). Research studies have deeply analyzed these systems and they converge into several aspects contributing to qualify the result of the matching process (e.g. *input, output, effort* …)[4] [5].

In a scalable context, a scalable matching system is able to match large and voluminous schemas (> 100 elements) [1] and has to be efficient and reliable providing both, fast execution and high-quality matching results. Further, it has to be adaptable to new execution environments, able to be easily installed, comparable with other matching tools, as well as extensible and flexible. Nowadays, several research work about scalable matching techniques are available, but, to the best of our knowledge, they lack of adequate methods to evaluate their operation [4][5][6].

Based on current approaches for software product quality, we propose in this paper an evaluation approach suitable to scalable matching tools features. With this approach, we attempt to provide matching experts with an useful method to asses, compose and improve scalable matching tools. Our approach is guided by an evaluation method and supported by a quality model adapted to the specific characteristics of scalable matching tools.

The rest of the paper is organized as follows: Section 2 introduces an overview of matching systems evaluation (schemas and ontology) and the adopted quality principles. Section 3 describes our approach. We show in Section 4, the results of the feasibility evaluation. Finally, we conclude and outline future work in Section 5.

II. OVERVIEW

According to current approaches tackling matching systems performance and studying the quality modeling used for software evaluation, we attempt to propose an evaluation method adapted to the characteristics of scalable matching tools. We detail these aspects as follows.

A. *Matching Systems evaluation*

According to [5] [6] [8] [9], there are several methods leaning to evaluate matching systems such as: *Benchmarks* and *comparative* and *application-based evaluations*. A benchmark allows researchers to evaluate their achievements not only in terms of performance, but also in terms of applicability in real world situations. In fact, approaches as [10] and [11] propose a performance evaluation based on a benchmark and on the execution time criterion over matching systems with large schemas (~844 elements). Others like [12] evaluate *efficacy* and *effectiveness* and propose some measures that combine both metrics in order to optimize the matching problem.

Considering comparative and application-based evaluation, several works were principally interested on proposing criteria adapted for schema and ontology matching. We distinguish two main criteria categories [5][9]: *Environment related* which is specified by data entry, evaluation conditions, etc. and *Matching related* considering matching techniques, auxiliary resources and the human effort. Others like [5][7] introduce criteria like *input, outputs, quality matching measures, effort* and use quality measures like *precision,* the *recall, F-measure* or *overall*. We notice that these criteria have been only used to evaluate systems in a small schema context (~60 elements). Only COMA, GLUE and Semit were tested with schemas containing maximum 300 elements.

Besides these approaches, [13] proposes the Quality of Matching (QoM) metric and several techniques to analyze in a qualitative (taxonomy) and quantitative way the QoM.

Although, this approach does not evaluate the quality of matching among different matching algorithms and therefore its applicability cannot be validated. Similarly, works like [14] [15] address ontology matching systems evaluation leading into OAEI test campaigns, using measures as *precision* and *recall* and evaluation criteria like *accuracy*, *complexity*, *incrementality* and *distinction capacity*.

As we can see, the most part of the existing evaluation methods are oriented to small scenarios (between ~60 and ~300 elements) and mainly focused on the use of measures like *precision* and *recall.* Such methods, generally lack of benchmarks adapted to the different nature, goals and operational principles of the matching tool, without taking into account the unavailability of matching tools (not all the ontology matching systems are available for comparison). Thus, a real need of an evaluation method adapted to scalable scenarios arises.

B. *Quality principles for tools evaluation*

Quality plays an important role in several actions performed to define, use and evolve particular systems (i.e. information and manufacturing systems). Currently, several evaluation approaches are provided in order to understand and improving software development [17][18][19][20][21]. In fact, there are few works that already address quality aspect in the scalable matching process (e.g. [13]).

Current approaches offer different *quality principles* (definition and modeling) to explicitly represent quality requirements and characteristics of a system [17]. Our perception of quality is related to software engineering domain [17][19][21]. Here, the evaluation of software product quality is a key factor and the objective of such an evaluation is to achieve the required quality of the product through the definition of quality requirements and their implementation, measurement of appropriate quality attributes and evaluation of resulting quality. In this context, the evaluation process is supported by a model integrating characteristics, sub-characteristics, measures and measurements. Characteristics such as *functionality, reliability, usability, efficiency, maintainability, portability* and sub-characteristics like *suitability, accuracy, and fault tolerance* among others.

We notice that quality principles in software engineering domain provide a structured way to evaluate tools, respecting the nature of matching systems. Thereby, we attempt to adapt such principles in order to qualify scalable matching tools.

III. QUALITY OF SCALABLE SCHEMA MATCHING SYSTEMS

Before introducing our approach, we clarify the terminology used in the rest of the paper. Inspired on [17] [18][19], we argue that the quality of a scalable matching system can be defined according to several quality characteristics (*scalability, maintainability, portability,* etc) (Figure 1). These characteristics are related to the behavioral features of the system and based on the specification of quality measures, measurement methods, functions and analysis models. Quality measures attempt to provide a quantitative representation of quality characteristics and quality measurements represent the set of operations determining the value of a measurement result.

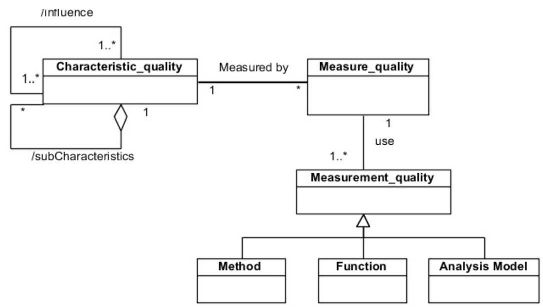

Figure 1. Quality model for Scalable Matching Tools Evaluation

Within this model, a quality characteristic is a specific feature of the matching tool and it can be characterized by a set of sub-characteristics. Moreover, quality characteristics can have a positive or negative influence over another quality characteristic. For each characteristic a set of different measures can be associated and for any given measure, one or several measurement methods, functions or analysis models can be applied.

A. *Evaluation method*

Despite the availability of many matching tools, there has been no standard method developed for comparing and evaluate them. As a result, and inspired on [19] and [21] we propose a method oriented to estimate the quality of scalable schema matching systems without interrupting its execution and able to be integrated on a benchmark level, on the existing approaches and helping to develop comparative or specific evaluation process (Figure 2).

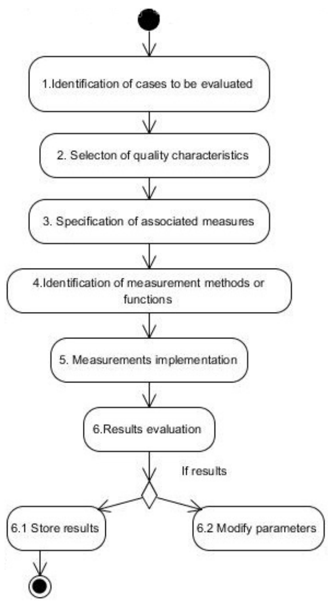

Figure 2. Evaluation method for Scalable Matching Tools

Our evaluation method integrates six phases. In phase 1, we identify the cases to be evaluated. Phase 2 implies the selection of quality characteristics related to each identified case. Phase 3, refers to the correlation of measures with the characteristics previously defined. In phase 4, we determine the methods, algorithms and equations capable to perform the required measures. Phase 5, implies experimentations using appropriate measures. The last phase is a decision making tread attempting to validate or not the acquired results. At this level, we distinguish two options: One considering results as validated (phase 6.1) and the other (phase 6.2) considering results as not validated. In the first one, the results will be used in a comparative evaluation to determine the reliability or performance of the scalable matching system. We can also store them for a further analysis or use them for system improvement associated to a benchmark. The other one is considering for example, a lot of noisy acquired results forcing to modify the evaluation parameters and repeat the required measures (Phase 5).

B. *Selection of quality characteristics*

The appropriate choice of quality characteristics ensures an optimal evaluation of scalable matching tools. Thus, inspired by the [19] and based on the particular requirements of scalable matching tools, we propose a set of quality characteristics judged as the most appropriated for this kind of systems (Figure 3).

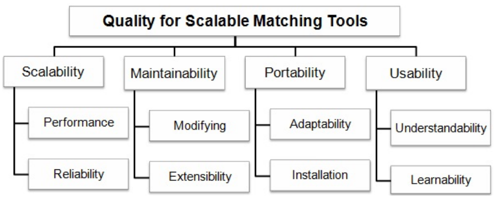

Figure 3. Quality characteristics for scalable matching tools

1) Scalability: This feature remains the most important quality aspect of any scalable matching tool. It corresponds to the capacity of a system to evolve [24]. We define scalability as "the system's ability to effectively handle large data by providing a fast response time and unaltered results when resources are appended". Scalability can be characterized in terms of *performance* and *reliability*. We consider performance as the relationship between the level of timeliness / system's response and the amount of resources used under certain conditions. Besides, a system is called reliable when the probability to satisfy some tasks over a given period of time corresponds to the specifications. In our context, reliability allows us to describe the level of suitability and completeness of a system.

2) Maintainability: Means the effort required to modify or extend a system. For a scalable matching tool it can be its ability to be extended at the addition of matchers, for example. Inspired on [19], we propose two characteristics: *modifying* and *extensibility*. Modifying means the capacity of a system to be modified in response to a fault or/and change on specifications and on the functional and nonfunctional requirements. Extensibility is the ability of the system or component to be extended in response to new requirements.

3) Portability: It is the capacity to migrate a system from one environment to another. This characteristic includes the ability to be adapted at different environments and with easy installation. We propose to describe this characteristic with *adaptability* and *installation*. Adaptability allows determining how a system can adapt itself to a changing environment. Installation allows specifying how a system can be installed in a given environment.

4) Usability: Refers to the effort required to employ a system by a defined or implicit set of users. It is described by *understandability* and *learnability* [19] representing the human effort expended to understanding the UI (User Interface) and the system management. *Learnability* is the capability of a system to enable the user to learn how to use it. With this sub-characteristic, we can estimate if the system is well documented and clear.

C. *Specification of associated measures*

In order to quantify the quality of a scalable matching tool, we have identified and propose several measures and measurement functions associated to the defined characteristics (Figure 3). In this paper, we particularly focus into *scalability* quality characteristic.

As we describe previously, *scalability* is characterized by *reliability* and *performance*. In order to quantify such sub-characteristics we propose several measures and measurement functions summarized in tables 1 and 2.

TABLE I. RELIABILITY ASSOCIATED MEASURES AND MEASUREMENTS

Reliability		
Measures	*Associated M.*	*Functions*
1) Relevancy	*Accuracy*	(1.1) $Accuracy = \frac{\lvert B\rvert}{\lvert B\rvert + \lvert C\rvert} = \frac{\lvert R \cap M\rvert}{\lvert M\rvert}$
	Noise	(1.2) $Noise = 1 - Accuracy$
2) Completeness	*Recall*	(2.1) $Recall = \frac{\lvert B\rvert}{\lvert A\rvert + \lvert B\rvert} = \frac{\lvert R \cap M\rvert}{\lvert R\rvert}$
	Silence	(2.2) $Silence = 1 - Recall$
	F-measure	(2.3) $F - measure = \frac{2 * \lvert B\rvert}{(\lvert A\rvert + \lvert B\rvert) + (\lvert B\rvert + \lvert C\rvert)} = \frac{2 * Accuracy * Recall}{Precision + Recall}$
3) Approximate similarity	*Accuracy w*	(3.1) $Accuracy_w = \frac{w(M,R)}{\lvert M\rvert}$
	Recall w	(3.2) $Recall_w = \frac{w(M,R)}{\lvert R\rvert}$
	Overlap	(3.3) $w(M,R) = \sum_{<m,r> \in C(M,R)} \sigma(m,r)$
	Compacteness	(3.4) $Compactness(M,R) = \frac{\lvert R\rvert}{\lvert M\rvert}$
4) Loss information	*Loss_information*	(4.1) $Information_{loss} = \frac{\lvert B - D\rvert}{\lvert B\rvert}$
5) Human effort	*Ed*	(5.1) $Ed = \frac{RPD - PPD}{PPD * 100}$
	Overall	(5.2) $Overall = 1 - \frac{\lvert A\rvert + \lvert C\rvert}{\lvert A\rvert + \lvert B\rvert} = \frac{\lvert B\rvert - \lvert C\rvert}{\lvert A\rvert + \lvert B\rvert} = Recall * \left(2 - \frac{1}{Precision}\right)$

Table 1 illustrates our proposition to measure reliability according to *relevancy* (1)*, completeness* (2)*, approximate similarity* (3)*, information loss* (4) *and human effort* (5)*.*

1) Relevancy. It is the capability of a scalable matching system to reject all false matches for a certain threshold. Associated measures are: *Accuracy* (1.1) and *Noise* (1.2). Accuracy is estimated using a set of true matches automatically founded (or true positive, B), the false matches proposed by the automatic matching (or false positive, C), a referential manually determined (R) and the Matching (M) returning by the system. Noise is a measure that allows quantifying the wrong matches provided by the system.

2) Completeness. It is the capability of a scalable matching system to return all the correct matches for a certain threshold. We propose to use *Recall* (2.1) and *Silence* (2.2) as associated measures. In order to estimate *Recall*, we take into consideration the matches that are not identified automatically (or false negatives, *A*) and the identified matches compared with a manually determined threshold (*R*) (determined by the expert as $|R|=|A|+|B|$). Silence is a complementary measure of Recall that allows determining the forgotten matches. We noticed that Accuracy and Recall measures are the most used measures to quantify matching results. However, a system comparison is more complex and thus, we propose to use other measure which represents a combination of both: *F-measure* (2.3).

3) Approximate similarity. The goal of this measurement is to extend the system evaluation to a decision making level and to the proximity between determined matches and the reference alignment ($|R \cap M|$). This involves determining "forgotten relatives" instead of an exact matching. We consider as related measures: *Accuracy* (3.1), the *Generalized recall* (3.2) and *Compactness* (3.4). Traditional measures do not distinguish between a matching (*M*) that can be very similar to the expected result (*R*) from another that is far enough from this result. Indeed, sometimes it is more probably to wonder if a proximity measure (*w*) of the matching was estimated, rather than if a particular match has been performed or not. Thus, we use an *overlap function* (3.3) to calculate the proximity of alignments. Such overlap take into account the best conformity of a certain matching regarding a threshold (*C(M,R)*) and compared to the best similarity measure between two matches ($\sigma(m,r)$). *Compactness* (3.4) is very useful measure in a scalable matching. This measure can be applied over a referential and a matching (*M*) and also between different types of schema matching. Such a measure also allows determining if the matches are close enough to be reused.

4) Loss of information. Aims to determine the loss of matches when a system uses particular techniques of fragmentation, partitioning, etc… To estimate such measure, we use the the function (4.1) based on the set of true matches automatically founded (or true positives, *B*) and the number of true matches determined by the use of decomposition approach (*D*).

5) Human effort. For the most part of the systems, the matching is a semi-automatic process requiring human intervention. We distinguish two cases of human effort: One creating the referential manually (this can take several days) and evaluated by *Effort deviation* measure (*Ed*) (5.1). In our case, the effort deviation depends on the time taken to complete the referential (*RPD - Real Person Days*) and the estimated time to achieve it (*PPD - Planned Person Days*). The other case is when editing the results generated by the automatic matchings. Such measure is estimated by the *Overall* [23] (5.2).

TABLE II. PERFORMANCE ASSOCIATED METRICS AND MEASURES

Performance		
Measures	*Associated M.*	*Functions*
6) Execution time	*Gain_time*	(6.1) $Gain_{exec} = \frac{exec\_time - exec\_time_o}{exec\_time}$
7) Memory space	*Space*	(7.1) *Measures of available system memory and system memory added at matching task.*

Table 2 resumes the measures related to *Performance*. Since the most part of the matching tools are not available for research, we propose two measures to estimate it: (6.1) *Execution time* and (7.1) *Memory space*.

6) Execution time. It is the response time of a system during the execution of different tasks (as schemas analysis, matching algorithms execution…). Such a measure is associated to the *gain_exec* (6.1) measurement which quantifies if the matching techniques have gain in time or the opposite. We estimate such gain considering an execution time (*exec_time*) without a decomposition approach and an execution time ($exec\_time_o$) using an optimization technique.

7) Memory space. Some matching systems meet several problems during the matching of large schemas often due to lack to storage and memory capacity. We consider interesting to measure either the memory allocated in the system as well as the memory added in order to carry out matching tasks.

IV. FESEABILITY EVALUATION

In this section, we attempt to demonstrate the feasibility of our approach and perform a comparative evaluation of two different scalable matching tools: PLASMA (Platform for LArge Schema Matching) [27] and COMA++ [1]. PLASMA and COMA++ share different features as matching and fragmentation of large schemas. In this paper we especially focus into two evaluation examples, including four of the seven defined metrics: *Relevancy*, *Completeness*, *Human effort* and *Execution time*. We provide some figures to illustrate the corresponding experimentation results.

A. PLASMA (Platform for Large Schema Matching) and COMA++

1) PLASMA. It is a scalable schema matching tool proposed in [25]. The architecture of PLASMA is deployed in three phases: *Pre-matching, Matching and Post-matching.* The pre-matching phase aims to decompose large schemas into smaller ones based on a holistic approach. The matching phase achieves matching between resulted schemas. The matching algorithm EXSMAL[26] considers linguistic and structural properties of schemas. The post-matching phase finds the correspondences between elements of schemas.

2) COMA++. It is a schema and ontology matching tool that offers a comprehensive and extensible library of individual matcher, which can be selected to perform a match operation [1]. Matchers included in this tool are more than fifteen, exploiting different kinds of schemas (e.g. simple string matchers as *Affix*, *Trigram*, *EditDistance*, etc., reusing oriented matchers and combined matchers) and auxiliary information.

B. Evaluation features

For each schema matching tool, we have deployed the proposed quality-oriented evaluation method in order to quantify the performance of these tools. We summarize the major characteristics of tested schemas in table 3.

TABLE III. CHARACTERISTICS OF E-BUSINESS SCHEMAS

Domain	No. Schemas	S/L *schema size (K bytes)	S/L* schema size (No. Elements)	Min/Max depth
XCBL	40	22/1130	22/7090	4/18
OAGIS	100	30/227	28/4480	5/13

1) PLASMA evaluation. Following our approach, evaluating PLASMA consists in the analysis of each deployment phase. According to our evaluation process, we identify different quality criteria according each phase:

- Pre-matching (Techniques): *schemas parser, mining algorithm, EXSMAL matching algorithm.*
- Matching (Auxiliary resources): *WordNet*
- Post-matching: *Quality measures and results.*

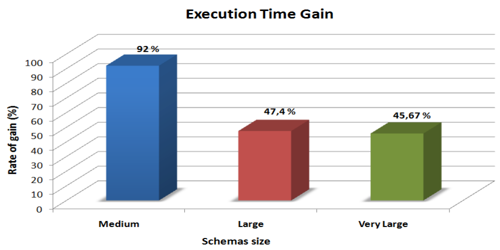

Figure 4. Evaluation of PLASMA matching

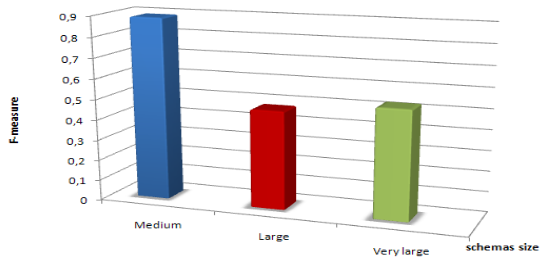

Figure 5. F-measure results

At this stage, we can notice the specificity of our method. In fact, using our quality perception we are able to specify all possible criteria at each level and evaluate them according to the corresponding characteristics, measures and measurements. For this experimentation, we have specially evaluated scalability at each phase of PLASMA. To this end, we used the different quality measures previously depicted in Section 3.B. For more details refers to [25].

Figure 6. Loss information results

In figure 4, we illustrate the results of evaluating execution time gain of matching phase. Figure 5 illustrates the relevancy and the completeness of matching. Then we use accuracy (or precision), recall and f-measure. In order to estimate the quality of the post-matching we have evaluated the reliability of the matching according to the loss of information metric (fig. 6).

2) COMA++ evaluation. Based on the traditional information retrieval metrics COMA++ and specially the fragmentation approach which is used for large scale matching, have been previously evaluated in [27]. However, we estimate that this evaluation lacks of elements to completely decide about COMA++'s scalability. In fact, we consider that using a fragmentation approach in a large scale context, leads to loss of information and real manual effort. Thus, we propose to extend this current evaluation technique with our approach and estimate the scalability of COMA++.

According to the architecture defined in [27], we identify the different evaluation phases of COMA++ and we propose complementary measures than those used in such approach. Such measures will be used to evaluate:

- *Parsing time* and *recovery rate* to evaluate fragmentation strategy, in schema manipulation module.
- *Matching performance* including performance of individual matchers.
- *Loss of information* and *manual effort* in the mapping manipulation.

For example, Figure7 illustrates the Loss of information at schema matching process in COMA++.

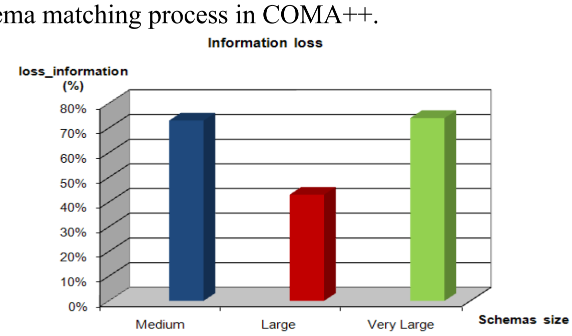

Figure 7. Loss information results

C. Concluding feseability evaluation

Our approach defines *scalability* in terms of *reliability* and *performance.* Then, based on these measures, we have applied our evaluation method on PLASMA and COMA++.

Accordingly, obtained results are very motivating and we can highlight various points. For example, traditional evaluation methods based on information retrieval measures are not capable to compare matching tools and are not well adapted to the large scale context. Also, the set of traditional measures are not able to quantify several aspects; for example the loss of information which is an important feature of scalable matching tool. Besides, our proposal can be applied to each phase or module of a matching system and then we are able to qualify the performance and reliability at each level of the system. Finally, in our approach the expert is a main actor, judging the performance of the process and the matching results.

V. CONCLUSIONS

In this paper, we have presented a quality-oriented approach to evaluate scalable matching tools. Our approach, proposes a quantitative evaluation according to different characteristics: *scalability, maintainability, portability,* and *usability.* We have described each one of them and presented their corresponding measures and measurement functions adapted to a scalable context. We have also presented some results derived from a feasibility evaluation of our approach over two matching tools: PLASMA and COMA++.

As we illustrate in this paper, our approach enable the comparison of scalable matching systems. But it also shows that evaluation methods could be continuously enhanced. For example, developing some referential for large schemas or to determine a single format that allows comparisons between several matching tools. We note also that the proposed measures and functions are currently suitable for scalable context, but they can be extended without any problem.

As future work, we intend to enhance our approach using quality levels and weights allow a qualitative evaluation of these tools. We attempt also to perform an implementation over current evaluation methods, especially those considering benchmarks. As we notice, the design of a benchmark for matching tools is challenging, especially due to the different characteristics of the tool.

REFERENCES

[1] Aumueller D., Do H. H., Massmann S., Rahm E.: Schema and Ontology Matching with COMA++. In: Ozcan Fatma. Proc. of the ACM SIGMOD Intl. Conf. on Management of Data, pp. 906-908, ACM Press, 2005.

[2] J Jian N., Hu W., Cheng G., Qu Y.: FalconAO: Aligning Ontologies with Falcon. In: Proc. of the K-CAP 2005 Workshop on Integrating Ontologies. CEUR Workshop Proceedings, 2005.

[3] Bernstein P., Melnik S., Petropoulos M.,Quix C.: Industrial-Strength Schema Matching. In:: ACM SIGMOD Record. 4, 38--43, 2004.

[4] E. Rahm, PA. Bernstein, "A survey of approaches to automatic schema matching" VLDB Journal, 10(4), pp. 334-350, 2001.

[5] Do H-H., Melnik S., Rahm E.: Comparison of schema Matching Evaluations. In: Web, Web-Services, and Database Systems, NODe Web and Database-Related Workshops, pp 221-237, Springer, 2002.

[6] Z. Bellahsene, A. Bonifati, F. Duchateau, Y. Velegrakis, "On Evaluating Schema Matching and Mapping". Iin: Schema Matching and Mapping, Data-Centric Systems and Applications, Part 2, Z. Bellahsene, A. Bonifati, E. Rahm eds. pp. 253-291, 2011.

[7] M. Hartung, J. F. Terwilliger, E. Rahm, "Recent Advances in Schema and Ontology Evolution", in Schema Matching and Mapping, Data-Centric Systems and Applications, Part 2, Z. Bellahsene, A. Bonifati, E. Rahm eds. pp. 149-190, 2011.

[8] Yatskevich M.: Preliminary Evaluation of schema matching systems. Tech. Report University of Trento, 2003.

[9] Euzenat J., Shvaiko P.: Ontology Matching. Heidelberg (DE): Springer, 341, 2007.

[10] Boukhebouze M., Rifaieh R., Benharkat A-N., Amghar Y.: Benchmarking XML-Schema Matching Algorithms for Improving Automated Tuning. In: Proc. of the IEEE/ACS Intl. Conf. on Computer Systems and Applications, pp. 917-925. IEEE, Jordan, 2007.

[11] Duchateau F., Bellahsene Z., Hunt E. : XBenchMatch: a Benchmark for XML Schema Matching Tools. In: Proc. of the 33rd Intl. Conf. on Very large Data Bases, pp. 1318-1321, 2007.

[12] Algergawy A., Schallehn E., Saake G.: Combining Effectiveness and Efficiency for Schema Matching Evaluation. In: Proc. of 1st Intl. Workshop on Model-Based Software and Data Integration, pp. 19-30,Springer, Germany , 2008.

[13] Tansalarak N.,Claypool K.T.: QoM: Qualitative and Quantitative Schema Match Measure. In: Proc. of the 22nd Intl. Conf. on Conceptual Modeling, pp. 31--44, Springer, Chicago, 2003.

[14] Shvaiko P., Euzenat J.: Ten challenges for ontology matching. In: Proc. of OTM Confederated International Conferences, pp. 1164-1182, 2008.

[15] Yatskevich M., Giunchiglia F., Avesani P.: A Large Scale Dataset for the Evaluation of Matching Systems. In: Posters of the 4th European Semantic Web Conference , 2007.

[16] Euzenat J.: Semantic Precision and Recall for Ontology Alignment Evaluation. In: Proceedings of the 20th Intl. Joint Conf. on Artificial Intelligence, pp. 348-353, Hyderabad, India, 2007.

[17] Garcia F., Bertoa M.F., Calero C., Vallecillo A. Ruiz F., Piattini M, Genero M.:Towards a consistent terminology for software measurement. In: Information & Software Tech. 48(8), pp. 631-644, 2006.

[18] Dromey, R.G.: Software Product Quality: theory, Model and Practice, Tech. Report, Software Quality Institute, Griffith University, Nathan, Brisbane ,1998.

[19] ISO, International Organization for standardization. ISO 25000::2005, Software product Quality Requirements and Evaluation (SQuaRE) Guide to SQuaRE. 2005.

[20] IEEE Computer Society, IEEE Standard 1061-1998, "IEEE Standard for a Software Quality Metrics Methodology", 1998.

[21] Kitchenham B.: DESMET: A method for evaluating Softwarre Engineering methods and tools. Technical Report TR96-09, U.K. 1996.

[22] Williams . G., Smith Connie U.: QSEM: Quantitative scalability evaluation method. In: Proc. of the 31th Intl. Computer Measurement group Conference, pp. 341--352 Computer Measurement group, Florida, 2005.

[23] Melnik S., Hector G-M.,Rahm E.: Similarity Flooding: A Versatile Graph Matching Algorithm and Its Application to Schema Matching. In:18th Intl. Conf. on Data Engineering, pp. 117--128, IEEE Computer Society San Jose, 2002.

[24] Sellami S.: Méthodologie de matching à large échelle pour des schemas XML. Thesis, INSA de Lyon , 2009.

[25] Sellami S., Benharkat A-N., Amghar Y.: Towards a more scalable schema matching: a novel approach. In: Intl. Journal of Distributed Systems and Technologies, vol 1, no 1, pp. 17--39 , IGI Publishing, 2010.

[26] Sellami S., Benharkat A-N., Amghar Y.,Flouvat F.: Improving real world schema matching with decomposition process. In: Proc. of the 12th Intl. Conf. on Enterprise Information Systems ICEIS, pp. 151--158 Portugal, 2010.

[27] Do H. H., Rahm E.: Matching large schemas: Approaches and evaluation. In: Information Systems journal, vol 32, n°6, pp. 857—885, 2007.