UNIVERSITY
OF TRENTO

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

38050 Povo – Trento (Italy), Via Sommarive 14
http://www.dit.unitn.it

SCHEMA-BASED SEMANTIC MATCHING: ALGORITHMS, A SYSTEM AND A TESTING METHODOLOGY

Mikalai Yatskevich

May 2005

Technical Report # DIT-05-047

International Graduate School in Information and Communication Technologies

Schema-based Semantic Matching: Algorithms, a System and a Testing Methodology

PhD Thesis Proposal

Mikalai Yatskevich
19*th* cycle
2004/2005

Dept. of Information and Communication Technology
University of Trento,
38050 Povo, Trento, Italy
yatskevi@dit.unitn.it

Schema-based Semantic Matching: Algorithms, a System and a Testing Methodology

Abstract. Schema/ontology/classification matching is a critical problem in many application domains, such as, schema/ontology/classification integration, data warehouses, e-commerce, web services coordination, Semantic Web, semantic query processing, etc. We think of Match as an operator which takes two graph-like structures and produces a mapping between semantically related nodes. Semantic matching is a novel approach where semantic correspondences are discovered by computing and returning as a result, the semantic information implicitly or explicitly codified in the labels of nodes and arcs. At present, the semantic matching approach is limited to the case of tree-like structures e.g., classifications, taxonomies, etc. The main focus of this PhD thesis, supervised by Prof. Fausto Giunchiglia is the development of the schema-based algorithm for semantic matching of tree-like structures; the development of the semantic matching system implementing the algorithm; and the development of the testing methodology allowing for a comprehensive evaluation of the semantic matching systems.

Keywords. Schema/ontology/classification matching, Semantic heterogeneity, Testing methodology.

1. Introduction

The progress of information and communication technologies, and in particular of the Web, has made a huge amount of heterogeneous information available. The number of different information resources is growing significantly, and therefore the problem of managing semantic heterogeneity is increasing. Many solutions to this problem include identifying terms in one information source that "match" terms in another information source. The applications can be viewed to refer to graph-like structures containing terms and their inter-relationships. These might be database schemas, classifications, taxonomies, or ontologies. The *Match* operator takes two graph-like structures and produces a mapping between the nodes of the graphs that correspond semantically to each other. We think of matching as the task of finding semantic correspondences between elements of two graph-like structures (e.g., classifications, conceptual hierarchies, database schemas or ontologies). Matching has been successfully applied to many well-known application domains, such as schema/ontology/classification integration, data warehouses, e-commerce, web services coordination, Semantic Web, semantic query processing, etc.

Semantic matching, as introduced in [2, 12] is based on the intuition that mappings should be calculated between the concepts (but not labels) assigned to nodes. Thus, for instance, two concepts can be equivalent; one can be more general than the other, and so on. This approach is based on two key notions, the notion of *concept of label*, and the notion of *concept at node*. They formalize the set of documents which one would classify under a label and under a node, respectively. As from [12], all previous approaches are classified as syntactic matching. These approaches, though implicitly or explicitly exploiting the semantic information codified in graphs, differ substantially from semantic matching approach in that, instead of computing semantic relations between nodes, they compute syntactic "similarity" coefficients between labels, in the [0,1] range [5, 16]. At present, the semantic matching approach is limited to the case of tree-like structures e.g., classifications, taxonomies, etc.

The main research goals within the PhD thesis are development of the schema-based algorithm for semantic matching of tree-like structures; development of the logical and physical architecture of the semantic matching system implementing the algorithm; development of the system and conduction of performance study for it; development of the methods for (semi) automatic generation of the matching problems and reference mappings acquisition; and development of the test cases collection admitting the comprehensive evaluation of the semantic matching systems.

The rest of the proposal is structured as follows: Section 2 expands more on the notions of matching problem, evaluation and testing methodology. Section 3 describes the existing approaches to matching and matching evaluation. In Section 4 the main objectives of the PhD thesis are defined. Section 5 concludes with the overview of work that has been done so far.

2. The matching problem and testing methodology

2.1 A motivating example

In order to motivate the matching problem, semantic matching approach, and illustrate one of the possible situations which can arise in the schema integration task let us use the two XML schemas A and B depicted on Figure 1. These schemas are taken from Yahoo and Standard business catalogues. Suppose that the task is to integrate these two schemas in order to allow interoperability between the information systems of the companies subscribed to the catalogues.

Fig. 1. Parts of Yahoo and Standard catalogues.

The first step in the schema integration is to identify candidates to be merged or to have taxonomic relationships under an integrated schema. This step refers to a process of schema matching. For example, $Computer\_Hardware_A$ can be assumed equivalent to $Computer\_Hardware_B$ and more general than $Personal\_Computers_B$. Hereafter the subscripts designate the schema (either A or B) from which the node is derived.

2.2 The matching problem

We assume that all the data and conceptual models (e.g., classifications, database schemas, taxonomies and ontologies) can be represented as graphs (see [12] for a detailed discussion). Therefore, the matching problem can be represented as extraction of graph-like structures from the data or conceptual models and matching the obtained graphs. This allows for the statement and solution of a *generic matching problem*, very much along the lines of what done in Cupid [16], and COMA [5].

We think of a *mapping element* as a 4-tuple $< ID_{ij}, n1_i, n2_j, R>$, $i=1,...,N1$; $j=1,...,N2$; where ID_{ij} is a unique identifier of the given mapping element; $n1_i$ is the *i-th* node of the first graph, $N1$ is the number of nodes in the first graph; $n2_j$ is the *j-th* node of the second graph, $N2$ is the number of nodes in the second graph; and R specifies a *similarity relation* of the given nodes. A *mapping* is a set of mapping elements. We think of *matching* as the process of discovering mappings between two graphs through the application of a matching algorithm. Matching approaches can be classified into *syntactic* and *semantic* depending on how mapping elements are computed and on the kind of similarity relation R used (see [12] for in depth discussion):

- In syntactic matching the key intuition is to find the syntactic (very often string based) similarity between the labels of nodes. Similarity relation R in this case is typically represented as a [0, 1] coefficient for example the *similarity* coefficients [16], [9]. Similarity coefficients usually measure the closeness between two elements linguistically and structurally. For example, the similarity between $Computer\_Storage\_Devices_A$ and $Data\_Storage\_Devices_B$ based on linguistical and structural analysis could be 0,63.
- Semantic matching is a new approach where semantic relations are computed between concepts (not between labels) at nodes. The possible semantic relations (R) are: *equivalence* (=); *more general* ($\supseteq$); *less general* ($\subseteq$); *mismatch* ($\perp$); *overlapping* ($\cap$). They are ordered according to decreasing binding strength, e.g., from the strongest (=) to the weakest ($\cap$). For example, as from Figure 1 $Computer\_Hardware_A$ is more general than $Large\_Scale\_Computers_B$. At present, the semantic matching approach is limited to the case of tree-like structures e.g., classifications, taxonomies, etc.

2.3 Matching evaluation

We think of evaluation in general as an assessment of performance or value of a system, process, product, or policy. When considering matching evaluation it is important to consider the requirements for any evaluation. As from [23] any evaluation requires:

- A *system* or its representation such as prototype, product, etc; together with a *process* (algorithm, simulation, etc).
- *Criteria* representing the objectives of the systems.
- *Measures* based on the criteria.
- *Measuring instruments* to register (or compute) the measures.
- *Methodology* for obtaining the measurements and conducting the evaluation (i.e., set of tools and methods applied in order to obtain the experimental results).

Let us review these requirements from the matching perspective. The *system* and *process* in this case include: the test collection and associated processing under given algorithms and procedures.

The major *criterion* exploited in the matching evaluation is *matching quality* which can be viewed as *relevance* (i.e., how relevant are the mappings produced by the matching system with respect to end user). The other criterion is *speed* (i.e., how fast the mappings are produced).

Well known in information retrieval *measures* of relevance (*Precision*, *Recall*, *Fallout*, and *F-measure*) were adapted to matching domain. Calculation of these measures is based on the comparison of the mapping produced by a matching system (*R*) with the reference mapping considered to be correct (*C*).

Precision is a correctness measure which varies from [0, 1]. It is calculated as

$$Precision = \frac{|C \cap R|}{|R|} \quad (1)$$

Recall is a completeness measure which varies from [0, 1]. It is calculated as

$$Recall = \frac{|C \cap R|}{|C|} \quad (2)$$

Fallout is a measure of incorrectness, which varies from [0, 1]. It is calculated as

$$Fallout = \frac{|R| - |C \cap R|}{|R|} \quad (3)$$

F-measure is a global measure of the matching quality. It varies from [0, 1] and calculated as a harmonic mean of *Precision* and *Recall*:

$$F\text{-}Measure = \frac{2*Recall*Precision}{Recall+Precision} \quad (4)$$

The other measure of the matching quality *Overall* varies in the [-1, 1] range; and calculated as the following combination of *Precision* and *Recall*:

$$Overall = Recall * \left(1 - \frac{1}{Precision}\right) \quad (5)$$

As a measure of *speed* execution *time* were taken.

In computation of the relevance based measures the experts play role of *measuring instruments*, since they define the relevance of the mappings. For execution *time* the clock are considered as a measuring instrument.

The testing *methodology* includes design, manner and techniques used to obtain and analyze the evaluation result. They also need to be evaluated for their validity, reliability and related criteria.

2.4 Testing methodology

From the matching perspective we distinguish the following categories (or levels) of the methodological issues:

- *Test cases collection*. This category includes the issues concerning the appropriateness of the test cases; their size and size of the collection; the methods and techniques of the test cases acquisition.
- *Results acquisition.* This category includes the issues concerning the methods of results and reference mappings acquisition, choice of the judges, and feedback.
- *Analysis and interpretation.* This category includes the issues concerning the techniques for the analysis of the results; the comparisons, in particular what comparisons are made and how; together with conclusions and generalizations of the experiment findings.

On the test cases collection level we distinguish between two categories of matching problems:

– The matching problems involving large size graphs with thousands and tenth of thousands of nodes. The good examples are WordNet [18], Google, Yahoo and Looksmart classification hierarchies.
– The matching problems involving small and medium graphs with tenth and hundreds of nodes. The structures of this size can be found in database schemas, ontologies, mail folders, etc.

These two categories of the matching problems differ significantly in the way of the reference mappings acquisition. In the first case the manual establishment of the reference mappings is hardly possible, since it is time consuming and error prone task. In the second case the problem of the subjectivity of the manually established reference mappings arises.

3. State of the art

3.1 Matching approaches

At present, there exist a line of the schema matching and ontology alignment systems (see [5], [16], [6], [21], [3], [10], [19], [17] for example). A good survey of the schema matching approaches up to 2001 is provided in [22]. In this survey the authors distinguish between several categories of matchers: *individual* and *combined*; *hybrid* and *composite*; *schema* and *instance based*; *element* and *structure level*. *Combined* matchers combine the results of *individual* matchers. *Composite* matchers combine the results of the several independently executed *individual* matchers, while *hybrid* ones exploit several approaches inside the matcher algorithm. *Individual* matchers exploit either *schema* or *instance level* information. There are two categories of the *schema based* matchers: *element* and *structure level*. *Element level* matchers consider only the information at the atomic level (e.g., the information contained in elements of the schemas), while *structure level* matchers consider also the information about the structural properties of the schemas.

The more recent work [24] improves [22] and elaborates in details *schema based* matching techniques. Consider Figure 2.

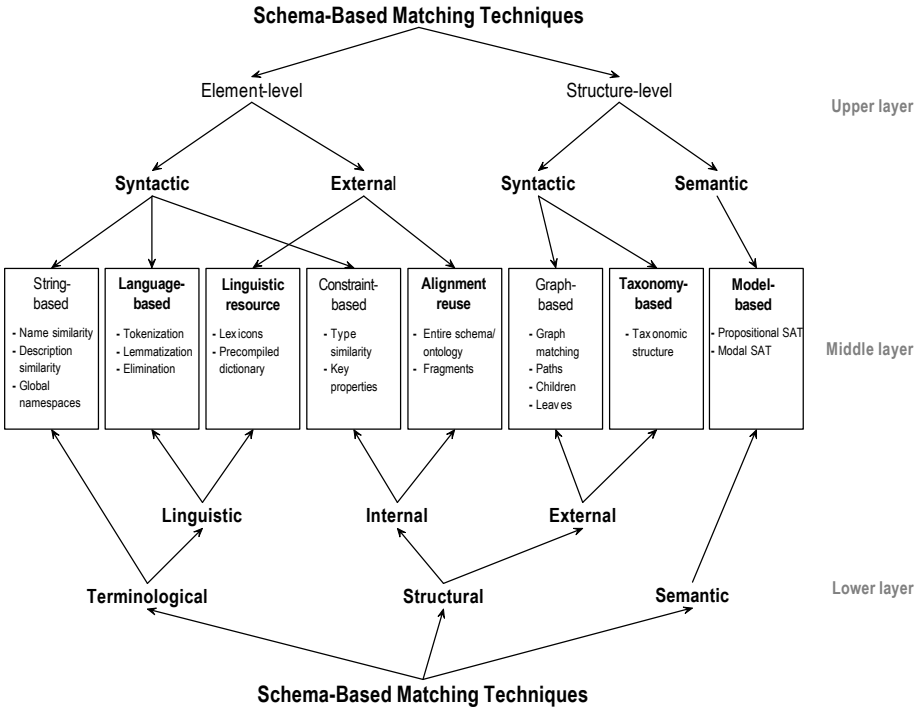

Fig. 2. The classification of the schema-based matching approaches presented in [24].

It presents the layered architecture, where the upper layer classify the matching approaches according to granularity of match and the way the matching techniques interpret the input information. Middle layer presents the classes of elementary techniques, while lower layer classify them according to the kinds of their input. The classification in Figure 2 can be read both in descending (focusing on how the techniques interpret the input information) and ascending (focusing on the kind of manipulated objects) manner.

According to [24] *syntactic* techniques interpret the input as a function of its sole structure using some clearly defined algorithm. *External* techniques exploit the external knowledge resources in order to interpret the input, while *semantic* exploit some formal semantics (e.g., model theoretic semantics). *Terminological* approaches work with strings, while *structural* exploit structures and *semantic* manipulate with models. Also *linguistic terminological* methods and techniques based on exploiting *internal* and *external structural* information are distinguished.

3.2 Matching systems

Let us review some of the state of the art schema-based matching systems and the efforts spent on their evaluation.

Anchor-PROMPT. The *Anchor-PROMPT* [21] (an extension of *PROMPT*, also formerly known as *SMART*) is an ontology merging and alignment tool. Its hybrid alignment algorithm takes as input two ontologies and a set of anchors-pairs of related terms, which are typically identified with the help of string-based techniques, or defined by a user. Then the algorithm refines them based on the ontology structure and users feedback. The quality of the mappings provided by *Anchor-PROMPT* has been initially evaluated on one pair of ontologies. *Precision* was calculated for various configurations of the system.

ASCO. ASCO [20] is an ontology matching system which combines in a hybrid manner the results obtained from element level techniques such as calculating the string distance, set similarity, etc. The results of element level techniques are the input for the structural algorithm, which exploits the ontology structure in order to propagate the similarities obtained in the element level matching phase. The propagation terminates after reaching the fix point. The system has been evaluated on one pair of real world ontologies with hundreds of nodes. *Precision, Recall* and *Overall* has been calculated.

Artemis. *Artemis* [3] was designed as a schema integration module of *MOMIS* [1] mediator system. *Artemis* exploits in a hybrid manner techniques at the element and structure-level. The system calculates the name, structural and global affinity coefficients by exploiting a common thesaurus. The common thesaurus presents a set of terminological and extensional relationships which depict intra and inter-schema knowledge about classes and attributes of the input schemas. A hierarchical clustering technique categorizes classes into groups exploiting global affinity coefficients. In the evaluation 7 databases were integrated in each of the two domains in the semi-automatic manner. Qualitative analysis of results was performed (similarity measures have not been calculated).

Cupid. The *Cupid* system [16] is a generic hybrid matcher exploiting element and structure level information. *Cupid* matching algorithm consists of three phases and operates only with tree-structures to which no-tree cases are reduced. The first phase (linguistic matching) computes linguistic similarity coefficients between schema element names (labels) based on morphological normalization, categorization, string-based techniques and a thesaurus look-up. The second phase (structural matching) computes structural similarity coefficients. The third phase (mapping generation) computes weighted similarity coefficients and generates final mappings by choosing pairs of schema elements with weighted similarity coefficients which are higher then a threshold. The system was comparatively evaluated against two other matching systems, *Dike* and *Artemis*. Qualitative analysis of the results has been performed.

COMA. The *COMA* system [5] is a generic schema matching tool. It exploits element and structure level techniques and combines the results in the composite way. *COMA* provides an extensible library of matching algorithms; a framework for combining obtained results, and a platform for the evaluation of the effectiveness of the different matchers. Matching library is extensible and contains 6 individual matchers, 5 hybrid matches and 1 reuse-oriented matcher. One of the distinct features of the *COMA* tool is the possibility of performing iterations in the matching process. It presumes interaction with a user which approves obtained matches and mismatches to gradually refine and improve the accuracy of match. *COMA* evaluation comprises 10 matching problems, among 5 XML schemas. The size of the schemas ranged from 40 to 145 elements. Evaluation consisted of over 12,000 test series (10 experiments in each). It helped to investigate the impact of different matchers and combination strategies on the match quality. The quality measures *Precision*, *Recall*, and *Overall* were determined for single experiments and then averaged over series. The best combination strategies were determined.

NOM and QOM. *NOM* [7] and its modification *QOM* [6] use COMA-like composite approach in order to combine the results obtained from element and structure level techniques. They are based on rules highly dependent from the knowledge explicitly codified in ontology such as subsumption and attribution relationships. *NOM* has been evaluated on 4 pairs of ontologies with hundreds of nodes. *Precision, Recall, F-Measure,* and *11-point measure* (*Precision* averaged on 11 points) has been calculated. *QOM* has been

evaluated on 3 pairs of ontologies with hundreds of nodes. The comparative evaluation against *NOM* and *Anchor-PROMT* has been performed. Time/quality trade of for the system has been demonstrated.

OLA. *OLA* [10] is an OWL-Lite ontologies matching tool. The system first calculates a set of distances (such as string distances) between the elements of the input ontologies. Than the distances is almost linearly aggregated into a system of linear equations. Afterwards, the fixed point algorithm is applied in order to find the solution that minimizes the distances. Finally, this solution is translated into a mapping according to predefined criteria. The matching problem in this hybrid approach is represented as an optimization problem. The system has been evaluated in EON contest [25]. *Precision*, *Recall* and *Fallout* have been calculated.

OMEN. *OMEN* [19] is an ontology matching tool which uses a hybrid approach. It takes as an input the existing mapping and uses a set of meta-rules in order to construct Bayesian network. The meta-rules capture the influence of the ontology structure and the semantics of ontology relations. The network is trained on the existing mappings and further used for obtaining the new ones. The system has been evaluated on the two ontologies from which the sets of sub graphs of size 11 and 19 nodes have been extracted. *Precision*, *Recall* and *F-measure* have been used as qualitative measures. The measures have been calculated for various strategies of the Bayesian network construction.

Similarity Flooding (SF). The SF [17] approach utilizes a hybrid matching algorithm based on the ideas of similarity propagation. Schemas in this approach are presented as directed labeled graphs. The technique obtains an initial mapping from string based element level matcher. Further the mapping is refined by the fix-point computation and filtered according to some predefined criteria. The system has been evaluated on 9 matching problems composed from XML and relational schemas. The biggest matching task was composed from the schemas with tenth of attributes. Matching accuracy (*Overall*) has been calculated for various filtering strategies and propagation coefficients.

3.3 Matching evaluation

Let us review the efforts spent on comparative evaluation of the matching systems from the light of the classifications presented in Sections 2.3 and 2.4.

In [4] 8 matching systems were compared on four criteria: *Output*, *Input*, *Quality measures* and (user) *Effort*. Since there have not been proposed the measures for estimation of *Output*, *Input* and *Effort*; these criteria were qualitatively compared among the systems. Since [4] presents the comparison of the matching system evaluations, the values of the quality measures were taken from the papers describing the systems. The best *Precision*, *Recall*, *F-Measure* and *Overall* reported by the authors of the systems have been compared.

In I3CON [26] 5 ontology alignment systems have been evaluated on the test bed consisting 8 ontology matching problems taken from various domains. The ontologies were taken from the web and matched against their modifications (e.g., adaptation to the concerned topic, language translation, etc.). The biggest matching problem was constructed from the ontologies with hundreds of classes. The reference mappings were produced by consensus of the external group of students. The set of tools for automation of the evaluation process has been provided. The qualitative measures (*Precision*, *Recall* and *F-Measure*) have been calculated for all the matching problems and compared among the systems.

In EON contest [25] 4 ontology alignment systems have been evaluated on the test bed consisting 20 matching problems. The initial ontology taken from bibliography domain was matched against its 16 modifications obtained in (semi) automatic way (e.g., flattened hierarchy, no instances, etc.) and 4 ontologies of the same domain developed by the different institutions. The ontologies were composed from tenth of classes. The reference mappings were known by participants in advance, what allowed them to tune their systems. The tools for mappings management and evaluation of the matching quality measures have been provided [8]. The qualitative measures (such as *Precision* and *Recall*) were calculated for all the matching problems and the analysis of results have been performed by the authors of the matching systems.

4. Objectives of the thesis work

The *primary objectives* of the thesis work are development of the algorithms for schema-based semantic matching of tree-like structures, the system implementing it, and a testing methodology allowing for a comprehensive evaluation of the semantic matching systems. As follows from Section 2, this is not a trivial task to perform, and it involves a number of different issues to be solved. In particular, the thesis work includes:

- *Development of the schema-based algorithm for semantic matching of tree-like structures.*
- *Development of the logical and physical architecture of the semantic matching system implementing the algorithm.*

– *Design and implementation of the semantic matching system based on the developed architecture.*
– *Performance study for the semantic matching algorithm.*
– *Development of the methods for (semi) automatic generation of the matching problems.*
– *Development of the methods for (semi) automatic reference mappings acquisition.*
– *Development of the test cases collection admitting the comprehensive evaluation of the semantic matching systems.*

5. What has been done so far

As the joint work with Prof. Giunchiglia and his research group, the semantic matching algorithm and the logical architecture of *S-Match*, the system implementing this algorithm, was developed (see [13] for more details).
The semantic matching algorithm is organized in the following four macro steps:
– *Step 1*: for all labels in the two trees, compute concepts of labels;
– *Step 2*: for all nodes in the two trees, compute concepts at nodes;
– *Step 3*: for all pairs of labels in the two trees, compute the semantic relations between concepts of labels;
– *Step 4*: for all pairs of nodes in the two trees, compute the semantic relations between concepts at nodes.

Step 1 and Step 2 are performed off line, while Step 3 and 4 are performed on line. In order to understand how the algorithm works, consider for instance the two trees depicted in Figure 3a.

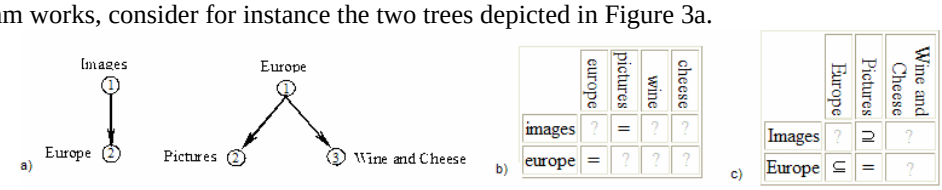

Fig. 3. (a): Two trees. (b): The matrix of relations between concepts of labels. (c): The matrix of relations between the concepts at nodes (matching result).

During *Step 1* we first tokenize the labels. For instance "*Wine and Cheese*" becomes <*Wine*, *and*, *Cheese*>. Then we lemmatize tokens. Thus for instance "*Images*" becomes "*image*". Then, an Oracle (at the moment we use WordNet 2.0) is queried in order to obtain the senses of the lemmatized tokens. Afterwards, these senses are attached to the atomic concepts. Finally, the complex concepts are built from the atomic ones. Thus, the concept of the label *Wine and Cheese* is computed as $C_{Wine\ and\ Cheese}$ = <*wine*, {$senses_{WN}$#4}>∧<*cheese*, {$senses_{WN}$#4}>, where <*cheese*, {$senses_{WN}$#4}> is taken to be the union of the four WordNet senses, and similarly for *wine*.

Step 2 takes into account the structural schema properties. The logical formula for a concept at a node is constructed most often as the conjunction of the concept of a label formulas in the concept path to the root [12]. For example, the concept C_2 for the node *Pictures* in Figure 3a is constructed as $C_2 = C_{Europe} \grave{U} C_{Pictures}$.

Element level semantic matchers are applied during *Step 3*. They determine the semantic relations holding between the atomic concepts of labels. For example, from WordNet we can derive that *image* and *picture* are synonyms. Therefore, $C_{Images} = C_{Pictures}$ can be inferred. Notice that *Image* and *Picture* have 8 and 11 senses in WordNet, respectively. In order to determine the relevant ones in the current context, sense filtering techniques are applied. The relations between the atomic concepts of labels for the trees depicted in Figure 3a are presented in Figure 3b.

Element level semantic matchers provide the input to the structure level matcher, which is applied in *Step 4*. This matcher produces as matching result the set of semantic relations between concepts at nodes (see Figure 3c for example). In this step the tree matching problem is reformulated into the set of node matching problems, one for each pair of nodes. Further, assuming, as background theory *context* [11], each node matching problem is reduced to a propositional validity problem.

S-Match, a system implementing semantic matching algorithm, was thought as a *platform* for semantic matching, namely a highly modular system where single components can be plugged, unplugged or suitably customized. The logical architecture of *S-Match*, is depicted in Figure 4.

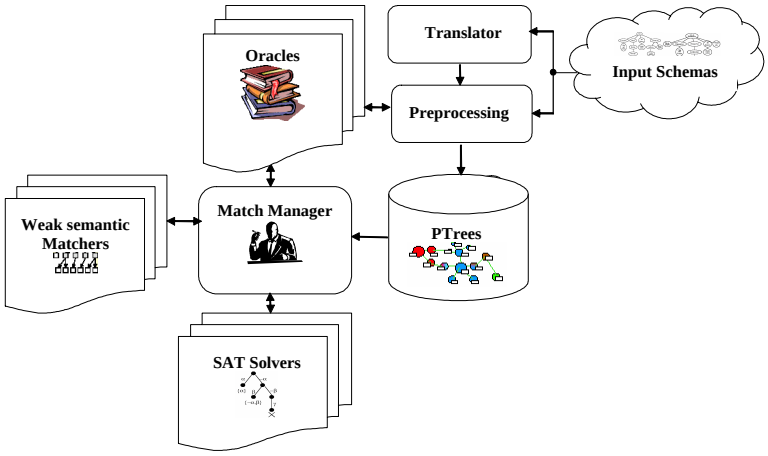

Fig. 4. Architecture of the S-Match platform.

Let us discuss it from a data flow perspective. The module taking input schemas does the *preprocessing*. It takes in input trees codified into a standard internal XML format. This internal format can be loaded from a file, manually edited or can be produced from an input format dependent *translator*. This module implements the preprocessing phase and produces, as output, enriched trees which contain concepts of labels and concepts of nodes. These enriched trees are stored in an internal storage (labeled *PTrees* in Figure 4) where they can be browsed, edited and manipulated. The preprocessing module has access to the set of *oracles* which provide the necessary *a priori* lexical and domain knowledge. In the current version WordNet 2.0 is the only Oracle we have. The *Matching Manager* coordinates the matching process using extensible libraries of *Oracles*; *weak semantic matchers*; and *SAT solvers*.

S-Match is implemented in Java. The system was comparatively evaluated against 3 state of the art matching systems on the test bed consisting 4 matching problems of relatively small size (the schemas up to tens of nodes). The quality (*Precision*, *Recall*, *Overall*, *F-Measure*) and time (*Time*) measures has been calculated (see [13] for more details). The results, though preliminary, look promising, in particular for what concerns *Precision* and *Recall*.

The analysis of the results obtained in the preliminary performance study identified the strong dependence of the matching quality measures from the element level semantic matchers. The improved version of the element level semantic matchers library, as described in [14] consist of 13 matchers classified into 3 major categories namely *string*, *sense* and *gloss based* matchers. All the matchers produce semantic relations as output. *String based* matchers compare two labels exploiting string comparison techniques. *Sense based* and *gloss based* matchers have two WordNet senses as an input. They exploit structural properties of WordNet hierarchies and gloss comparison techniques, respectively.

The analysis of the results obtained in the preliminary performance study identified the strong dependence of execution time from the reasoning techniques exploited by structure level matcher. The set of optimizations to the basic version of the structure level algorithm have been developed. Particularly, the linear time algorithm for solving certain propositional unsatisfiability problems arising in the semantic matching process was presented. The optimizations significantly improve the performance of *S-Match* on the trees evaluated so far (see [15] for more details).

6. Future work

The results described in the previous section are encouraging, but leave space for far more investigations.

The first version of the semantic matching algorithm has several shortcomings. In particular we distinguish between two lines of the semantic matching algorithm evolution:

– *Improvement of the matching quality.* As the preliminary evaluation results show, lack of information provided by element level semantic matchers crucially influence on the matching quality. In order to cope with this problem the new methods of element level semantic matching needs to be developed. One of the possible directions of the semantic matching algorithm evolution is the development of the new corpus and gloss based element level semantic matching techniques. As from [14], the corpus based techniques are highly dependent from the size and relevance of the corpus they exploit. The promising direction is exploiting web search engines as sources of relevant corpuses for the particular matching tasks. The new

gloss based techniques will consider the grammatical structure of the glosses. The other promising direction is exploiting of the glosses parsed into logical formulas. In this approach element level matching is performed by a reasoning procedure which determines the semantic relation holding between the glosses, similarly to *S-Match* structure level algorithm.

Incorrect or incomplete input information also significantly influence on the quality of the results provided by the semantic matching algorithm. For example, the labels of nodes can contain acronyms, abbreviations, etc. This means that the algorithm needs to be extended in order to deal with incorrect or incomplete input information (i.e., the robustness of the algorithm needs to be improved). Possible ways of the algorithm evolution includes improvement of the linguistic preprocessing techniques and development of ad hoc techniques for dealing with incorrectness and incompleteness in the input information.

– *Improvement of the matching efficiency.* The work on improvement of the matching efficiency necessary requires identification of the bottlenecks in the current version of *S-Match* and providing suitable modifications to the matching algorithm. For example, the new reasoning techniques can be used in order to improve the performance of the structure level matcher. The other optimizations include efficient algorithms for element level semantic matchers and look forward techniques for determining of the irrelevant to the particular matching task parts of the trees.

The changes in the semantic matching algorithm require suitable corrections in the logical architecture. The architecture also need to be adjusted in order to allow possibility of reuse *S-Match* platform and its parts in the other application domains such as documents classification, Semantic Web browsing, etc.

The implementation of the changes in semantic matching algorithm and architecture necessarily require extension of the *S-Match* system. Improvements in linguistic preprocessing engine, new element and structure level matchers need to be implemented.

The main goal of the performance study is to identify the dependence of a wide range of indicators from a variety of factors which influence on the matching process. In particular, dependence of the execution time and matching quality measures from the size of the trees, structural patterns of the tree structure, and the structure of the labels of nodes will be determined. The experiments will help in identification of shortcomings and bottlenecks in the algorithm and system. They will also provide the clues to the ways of the system evolution. A special attention should be given to identification of the “hard” and “easy” matching problems and the factors which influence on their complexity.

(Semi) automatic generation of the matching problems can be performed by extraction of trees from large size graph-like structures (such as WordNet). In this case the labels of the trees are human readable and the matching problems contain given number of semantic relations holding between the elements of the trees.

A special attention should be given to the development of techniques for (semi) automatic reference mappings acquisition from the artificially generated matching problems and from instance level information. Latter is especially useful for the matching problems composed from large size tree-like structures, when the acquisition of the reference mappings exploiting the other methods is hardly possible. For example, the number of the same URIs classified under the nodes in Google and Yahoo classifications can give evidence to the reference mapping discovery.

Acknowledgments

I would like to thank Fausto Giunchiglia, my supervisor, for many lessons on how to do research and write articles, for his invaluable support and never-ending patience in guiding my entrance into AI domain and life in general.

List of publications

1. F. Giunchiglia, M. Yatskevich, E. Giunchiglia. Efficient Semantic Matching. Submitted to *ESWC’05*. Long version in Technical Report DIT-04-110, University of Trento, 2004.
2. F. Giunchiglia and M. Yatskevich. Element level semantic matching. In *Proceedings of Meaning Coordination and Negotiation workshop* at ISWC, 2004.
3. F. Giunchiglia, P. Shvaiko, M. Yatskevich. S-Match: An algorithm and an implementation of semantic matching. In *Proceedings of ESWS'04*., pages 61–75, 2004.

References

[1] S. Bergamaschi, S. Castano, M. Vincini. Semantic Integration of Semistructured and Structured Data Sources. In *SIGMOD Record*, 28(1): 54-59, 1999.

[2] P. Bouquet, L. Serafini, S. Zanobini. Semantic Coordination: A new approach and an application. In *Proceedings of ISWC*, 2003.

[3] S. Castano, V. De Antonellis, and S. De Capitani di Vimercati. Global viewing of heterogeneous data sources. In *IEEE Transactions on Knowledge and Data Engineering*, number 13(2), pages 277–297, 2001.

[4] H. Do, S. Melnik, E. Rahm. Comparison of schema matching evaluations. In *Proceedings of workshop on Web and Databases*, 2002.

[5] H. Do, E. Rahm. COMA - A system for Flexible Combination of Schema Matching Approaches, In *Proceedings of VLDB*, 2002.

[6] M. Ehrig and S. Staab. QOM: Quick ontology mapping. In *Proceedings of ISWC*, 2004.

[7] M. Ehrig and Y. Sure. Ontology mapping - an integrated approach. In *Proceedings of ESWS*, pages 76–91, 2004.

[8] J. Euzenat, An API for ontology alignment, In *Proceedings of ISWC*, pages 698-712, 2004.

[9] J. Euzenat, P. Valtchev. An integrative proximity measure for ontology alignment. In *Proceedings of Semantic Integration workshop at ISWC*, 2003.

[10] J. Euzenat, P. Valtchev, Similarity-based ontology alignment in OWL-lite, In *Proceedings of ECAI 2004*, pages 333–337.

[11] F. Giunchiglia. Contextual reasoning. *Epistemologia, special issue on "I Linguaggi e le Macchine"*, vol. XVI: 345-364, 1993.

[12] F. Giunchiglia, P. Shvaiko. Semantic Matching. In *The Knowledge Engineering Review Journa*l, 18(3) 2003.

[13] F. Giunchiglia, P. Shvaiko, M. Yatskevich. S-Match: An algorithm and an implementation of semantic matching. In *Proceedings of ESWS*, pages 61–75, 2004.

[14] F. Giunchiglia and M. Yatskevich. Element level semantic matching. In *Proceedings of Meaning Coordination and Negotiation workshop at ISWC, 2004*.

[15] F. Giunchiglia, M. Yatskevich, E. Giunchiglia. Efficient Semantic Matching. Submitted to ESWC'05. Long version in DIT Technical Report DIT-04-110, 2004.

[16] J. Madhavan, P. Bernstein, E. Rahm. Generic Schema Matching with Cupid. In *Proceedings of VLDB 2001*.

[17] S. Melnik, H. Garcia-Molina, E. Rahm. Similarity Flooding: A Versatile Graph Matching Algorithm. In *Proceedings of ICDE*, 2002 117-128.

[18] A. Miller, Wordnet: A lexical database for English. In *Communications of the ACM*, 38(11): 39-41, 1995.

[19] P. Mitra, N. Noy, A. Jaiswal. OMEN: A Probabilistic Ontology Mapping Tool In *Proceedings of Workshop on Meaning coordination and negotiation at ISWC-2004*, Hisroshima, Japan.

[20] B. Thanh Le, R. Dieng-Kuntz, F. Gandon. In *proceedings of 6th International Conference on Enterprise Information Systems* Universidade Portucalense, Porto - Portugal, 14-17, April 2004

[21] N. Noy, M. Musen. Anchor-PROMPT: Using Non-Local Context for Semantic Matching. In *Proceedings of IJCAI* workshop *on Ontologies and Information Sharing*, (2001) 63-70.

[22] E. Rahm, P.Bernstein. A survey of approaches to automatic schema matching. In *VLDB Journal*, 10(4): 334-350, 2001.

[23] T. Saracevic. Evaluation of evaluation in information retrieval. In *Proceedings of the 18th ACM SIGIR conference*.

[24] P. Shvaiko and J. Euzenat. A Survey of Schema-based Matching Approaches. Submitted to *Journal of Data Semantics*, 2004. Long version in DIT Technical Report DIT-04-87, 2004.

[25] Y. Sure, O. Corcho, J. Euzenat, T. Hughes, editors. *Proceedings of the 3rd Evaluation of Ontology-based tools* (EON), 2004.

[26] http://www.atl.external.lmco.com/projects/ontology/i3con.html