



UNIVERSITY
OF TRENTO

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

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

AN ALGORITHM FOR MATCHING CONTEXTUALIZED
SCHEMAS VIA SAT

Luciano Serafini, Paolo Bouquet, Bernardo Magnini
and Stefano Zanobini

January 2003

Technical Report # DIT-03-003

An algorithm for matching contextualized schemas via SAT

Luciano Serafini[†] Paolo Bouquet^{† ‡} Bernardo Magnini[†] Stefano Zanobini[‡]
serafini@itc, {bouquet,zanobini}@dit.unitn.it

[†]ITC-irsT, Via Sommarive 18 – Povo, 38050 Trento, Italy

[‡]DIT University of Trento, Via Sommarive 12 – Povo, 38050 Trento, Italy

Abstract

The development of more and more complex distributed applications over large networks of computers has raised the problem of *semantic interoperability* across autonomous applications. In this paper we propose an algorithm, called CTX-MATCH, for discovering semantic relations between concepts belonging to heterogeneous and autonomously developed semantic schemas. The most significant innovations of the algorithm, which is theoretically founded on a well-known theory of contextual reasoning in AI, are that (i) the problem of finding relationships between concepts in different schemas is encoded as a problem of logical satisfiability (and so mappings have a well-defined semantic); and (ii) the way linguistic and domain knowledge is used to build the SAT problem. In this paper, we are mainly focused on the first aspect. The algorithm has been implemented as part of a peer-to-peer system for Distributed Knowledge Management, and tested on significant cases.

1 Introduction

The development of more and more complex distributed applications over large networks of computers has created a whole new class of conceptual, technical, and organizational problems. Among them, one of the most challenging one is the problem of *semantic interoperability*, namely the problem of allowing the exchange meaningful information/knowledge across applications which (i) use autonomously developed conceptualizations of their domain, and (ii) need to collaborate to achieve their users' goals.

Two are the main approaches proposed for solving the problem of semantic interoperability. The first is based on the availability of shared semantic structures (e.g., ontologies, global schemas) onto which local representations can be totally or partially mapped. The second is based on the creation of a global representation which integrates local representations. Both approaches do not seem suitable in scenarios where: (i) local representations are updated and changed very frequently, (ii) each local representation is managed in full autonomy w.r.t. the other ones, (iii) local representations may

appear and disappear at any time, (iv) the discovery of semantic relation across different representations can be driven by a user's query, and thus cannot be computed beforehand (runtime discovery) nor take advantage of human intervention (automatic discovery).

In this paper we propose an algorithm for runtime and automatic discovery of semantic relations across local representations. The most significant innovations of the algorithm, which is theoretically founded on a well-known theory of contextual reasoning in AI [Ghidini and Giunchiglia, 2001; Benerecetti *et al.*, 2000], are that (i) the problem of finding relationships between concepts in different schemas is encoded as a problem of logical satisfiability (and therefore mappings have a well-defined semantic); and (ii) the way linguistic and domain knowledge is used to build the SAT problem.

First, we characterize the scenarios that motivate our approach to schema matching, and explain why we use the theory of context as a theoretical background of the algorithm. Then, we describe the macro-blocks of the algorithm, namely semantic explicitation and context mapping via SAT. Finally, we briefly compare our algorithm with some other proposals in the literature.

2 Motivating scenarios

The work on the algorithm was originally motivated by a research on Distributed Knowledge Management [Bonifacio *et al.*, 2002b], namely a distributed approach to managing corporate knowledge in which users (or groups of users, e.g. communities) are allowed to organize their knowledge using autonomously developed schemas (e.g., directories, taxonomies, corporate ontologies), and are then supported in finding relevant knowledge in other local schemas available in the corporate network.

In this scenario, the algorithm we present aims at solving the following problem. Let s (the *source schema*) and t (the *target schema*) be two autonomous schemas that different users (or groups) use to organize and access a local body of data. Given a concept k_s in s , and a concept k_t in t , what is the semantic relations between k_s and k_t ? For example, are the two concepts equivalent? Or one is more (less) general than the other one? In addressing this problem, it is assumed that the basic elements of each schema are described using words and phrases from natural language (e.g., English, Italian); this reflects the intuition that schemas encode a lot of

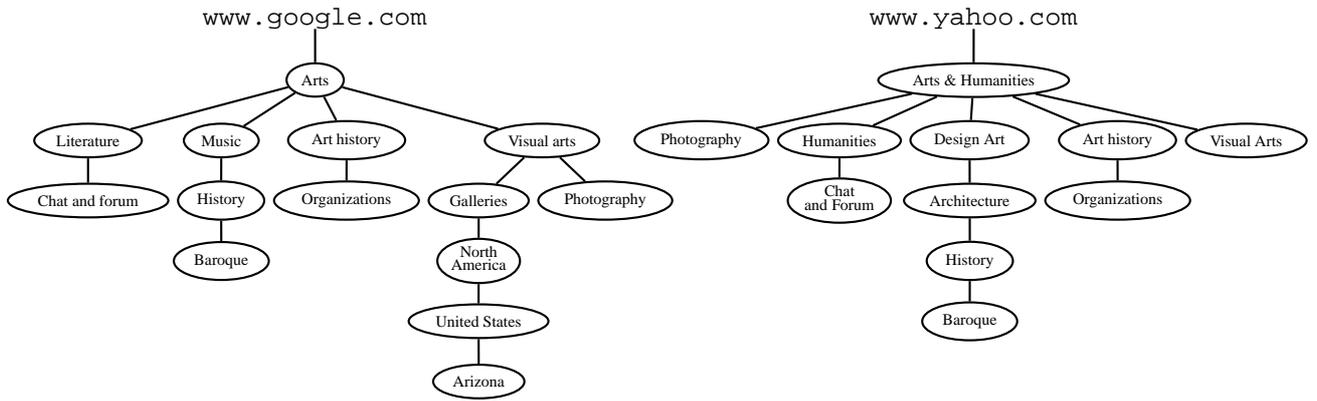


Figure 1: Examples of concept hierarchies (source: Google and Yahoo)

implicit knowledge, which can be made explicit only if one has access to the meaning of the words that people use to denote concepts in the schema.

Scenarios with similar features can be found in other important application domains, such as the semantic web (where each site can have a semantic description of its contents and services), marketplaces (where every participating company may have a different catalog, and every marketplace may adopt a different standard for cataloging products); search engines (some of them, e.g. the Google and the Yahoo, provide heterogeneous classifications of web pages in web directories); the file system on the PCs of different users (where each user stores documents in different directory structures). So the class of applications in which our algorithm can be applied is quite broad.

3 Local schemas as contexts

In many interesting applications, schemas are directed graphs, whose nodes and edges are labeled with terms or phrases from natural language. A typical example is depicted in Figure 1, whose structures are taken from the Google and Yahoo directories. In this section, we briefly argue why we interpret these schemas as contexts in the sense of [Benerecetti *et al.*, 2000] (see [Ghidini and Giunchiglia, 2001] for a formalization).

In schemas like the ones in the figure, the meaning of a label depends not only on its linguistic meaning (what a dictionary or thesaurus would say about that word or phrase), but also on the context in which it occurs: first, it depends on the position in the schema (e.g., the documents we as humans expect to find under the concept labeled *Baroque* in the two structures in Figure 1 are quite different, even if the label is the same, and is used in the same linguistic sense); second, it depends on background knowledge about the schema itself (e.g., that there are chat and forums about literature helps in understanding the implicit relation between these two concepts in the left hand side schema). These contextual aspects of meaning are distinct (though related) to purely linguistic meaning, and we want to take them into account in our algorithm.

To this end, the algorithm we present in this paper is applied to contexts rather than to schemas directly. In [Benerecetti *et al.*, 2000], a context is viewed as a box, whose content is an explicit (partial, approximate) representation of some domain, and whose boundaries are defined by a collection of assumptions which hold about the explicit representation. The notion of context we use in this paper is a special case of the notion above. A context is defined as a pair $c = \langle R_c, A_c \rangle$, where:

1. R_c is a graph, whose nodes and edges can be labeled with expressions from natural language;
2. A_c is a collection of explicit assumptions, namely attributes (parameter/value pairs) that provide meta-information about the content of the context.

In the current version of the algorithm, we restrict ourselves to the case in which R_c is a concept hierarchy (see Def. 3.1), and the explicit assumptions A_c are only three: the *id* of the natural language in which labels are expressed (e.g., English, Italian), the reference structure R_c of the explicit representation (the only accepted value, at the moment, is “concept hierarchy”, but in general other values will be allowed, e.g., taxonomy, ontology, semantic network, frame), and the domain theory (see below for an explanation of this parameter). Their role will become apparent in the description of the algorithm.

A concept hierarchy is defined as follows:

Definition 3.1 (Concept hierarchy). A concept hierarchy is a triple $H = \langle K, E, l \rangle$ where K is a finite set of nodes, E is a set of arcs on K , such that $\langle K, E \rangle$ is a rooted tree, and l is a function from $K \cup E$ to a set L of strings.

Definition 3.2 (Hierarchical classification). A hierarchical classification of a set of documents D in a concept hierarchy $H = \langle K, E, l \rangle$ is a function $\mu : K \rightarrow 2^D$.

μ satisfies the following *specificity principle*: a user classifies a document d under a concept k , if d is about k (according to the user) and there isn’t a more specific concept k' under which d could be classified¹.

¹See Yahoo instruction for “Finding an appropriate Category” at <http://docs.yahoo.com/info/suggest/appropriate.html>.

Mappings between contexts are defined as follows:

Definition 3.3 (Mapping function). A mapping function M from $H = \langle K, E, l \rangle$ to $H' = \langle K', E', l' \rangle$ is a function $M : K \times K' \rightarrow rel$, where rel is set of symbols, called the possible mappings.

The set rel of possible mappings we consider in this paper contains the following: $k_s \supseteq k_t$, for k_s is more general than k_t ; $k_s \subsetneq k_t$ for k_s is less general than k_t ; $k_s \overset{*}{\rightarrow} k_t$ for k_s is compatible with k_t ; $k_s \perp k_t$ for k_s is disjoint from k_t ; $k_s \equiv k_t$ for k_s is equivalent to k_t . The formal semantics of these expressions is given in terms of compatibility between document classifications of H_s and H_t :

Definition 3.4. A mapping function M from H_s to H_t is extensionally correct with respect to two hierarchical classifications μ_s and μ_t of the same set of documents D in H_s and H_t , respectively, if the following conditions hold for any $k_s \in K_s$ and $k_t \in K_t$:

$$\begin{aligned} k_s \supseteq k_t &\Rightarrow \mu_s(k_s \downarrow) \supseteq \mu_t(k_t \downarrow) \\ k_s \subsetneq k_t &\Rightarrow \mu_s(k_s \downarrow) \subsetneq \mu_t(k_t \downarrow) \\ k_s \perp k_t &\Rightarrow \mu_s(k_s \downarrow) \cap \mu_t(k_t \downarrow) = \emptyset \\ k_s \equiv k_t &\Rightarrow \mu_s(k_s \downarrow) = \mu_t(k_t \downarrow) \\ k_s \overset{*}{\rightarrow} k_t &\Rightarrow \mu_s(k_s \downarrow) \cap \mu_t(k_t \downarrow) \neq \emptyset \end{aligned}$$

where $\mu(c \downarrow)$ is the union of $\mu(d)$ for any d in the subtree rooted at c .

The semantics in Definition 3.4 is a particular case of relation between contexts (i.e., compatibility relation) defined in the Local Models Semantics of [Ghidini and Giunchiglia, 2001; Borgida and Serafini, 2002]. The algorithm we propose can be viewed as a first attempt of automatically discovering compatibility relations across contexts.

4 The Matching Algorithm

The algorithm has two main phases:

Semantic explicitation In the schema level, a lot of information is implicit in the labels, and in the structure. The objective of this first phase is to make it as explicit as possible by associating to each node (and edge) k a logical formula $w(k)$ that encodes this information. Intuitively, $w(k)$ is an approximation of the human interpretation.

Semantic comparison We encode the problem of finding mappings between two concepts k and k' , whose explicit meaning is $w(k)$ and $w(k')$, into a problem of satisfiability, which is then solved by a SAT solver in a logic W (i.e., the logic in which $w(c)$ and $w(c')$ are expressed). Domain knowledge is also encoded as a set of formulas of W .

Since here we are mainly focussed on the second phase, we only provide a short description of semantic explicitation (details can be found in [Magnini *et al.*, 2002a]), and then move to the SAT encoding.

4.1 Semantic explicitation

The goal of the first phase is to make explicit all the semantic information which can be fruitfully used to define the SAT problem in a rich way. The main intuition is that any schema is interpreted (by its users) using two main sources of information: lexical information, which tells us that a word (or a phrase) can have multiple senses, synonyms, and so on; and a background theory, which provides extra-linguistic information about the concepts in the schema, and about their relations. For example, lexical information about the word “Arizona” tells us that it can mean “a state in southwestern United States” or a “glossy snake”. The fact that snakes are animals (reptiles), that snakes are poisonous, and so can be very dangerous, and so on, are part of a background theory which one has in mind when using the word “Arizona” to mean a snake². In the version of the algorithm we present here, we use WORDNET as a source both of lexical and background information about the labels in the schema. However, we’d like to stress the fact that the algorithm does not depend on the choice of any particular dictionary or theory (i.e., does not depend on WORDNET). Moreover, we do not assume that the same dictionary and background theory are used to explicit the semantic of the two contexts to be matched.

Semantic explicitation is made in two main steps: *linguistic interpretation* and *contextualization*.

Linguistic interpretation

Let $H = \langle K, E, l \rangle$ be a concept hierarchy and L_H the set of labels associated to the nodes and edges of a hierarchy H by the function l . In this phase we associate to each label $s \in L_H$ a logical formula representing the interpretation of that label w.r.t. the background theory we use.

Definition 4.1 (Label interpretation). Given a logic W , a label interpretation in W is a function $I : L_H \rightarrow wff(W)$, where $wff(W)$ is the set of well formed formulas of W .

The choice of W depends on the external assumptions of the context containing H . For concept hierarchies, we adopted a description logic W with \sqcup , \sqcap and \neg , whose primitive concepts are the synsets of WORDNET that we associate to each label (with a suitable interpretation of conjunctions, disjunctions, multi-words, punctuation, and parenthesis). For example, WORDNET provides 2 senses for the label *Arizona* in Figure 1, denoted by #1 and #2; in this case, the output of the linguistic analysis is the following formula in W : $Arizona\#1 \sqcup Arizona\#2$

Contextualization

Linguistic analysis of labels is definitely not enough. The phase of contextualization aims at pruning or enriching the synsets associated to a label in the previous phase by using the context in which this label occurs. In particular, we introduce the concept of *focus* of a concept k , namely the smallest subset of H which we need to consider to determine the

²We are not saying here that there is only one background theory. On the contrary, theories tend to differ a lot from individual to individual, and this is part of the reason why communication can fail. What we are saying is that, to understand what “Arizona” means in a schema (such as the concept hierarchy in the left hand side of Figure 1), one must have a theory in mind.

meaning of k . What is in the focus of a concept depends on the structure of the explicit representation. For concept hierarchies, we use the following definition:

Definition 4.2 (Focus). *The focus of a concept $k \in K$ in a concept hierarchy $H = \langle K, E, l \rangle$, is a finite concept hierarchy $f(k, H) = \langle K', E', l' \rangle$ such that: $K' \subseteq K$ contains k , its ancestors, and their direct descendants; $E' \subseteq E$ is the set of edges between the concepts of K' ; l' is the restriction of l on K' .*

The *contextualization* of the interpretation of concept k of a context c is formula $w(k)$, called *contextualized interpretation* of k , which is computed by combining the linguistic interpretations associated to each concept h in the focus of k . The two main operations performed to compute $w(k)$ are sense filtering and sense composition.

Sense filtering uses NL techniques to discard synsets that are not likely to be correct for a label in a given focus. For example, the sense of *Arizona* as a snake can be discarded as it does not bear any explicit relation with the synsets of the other labels in the focus (e.g., with the synsets of *United States*), whereas it bears a part-of relation with *United States*#1 (analogously, we can remove synsets of *United States*).

Sense composition enriches the meaning of a concept in a context by combining in linguistic interpretation with structural information and background theory. For concept hierarchies, we adopted the default rule that the contextual meaning of a concept k is formalized as the conjunction of the senses associated to all its ancestors. Furthermore, some interesting exceptions are handled. An example: in the Yahoo Directory, *Visual arts* and *Photography* are sibling nodes under *Arts & Humanities*; since in WORDNET photography is in a is-a relationship with visual art, the node *Visual arts* is re-interpreted as visual arts minus photography, and is then formalized in description logic as: $\text{visual_art\#1} \sqcup \neg \text{photography\#1}$

4.2 Computing relations between concepts via SAT

In the second phase of the algorithm, the problem of discovering the relationship between a concept k in a context c and a concept k' in a context c' is reduced to the problem of checking, via SAT, a set of logical relations between the formulas $w(k)$ and $w(k')$ associated to k and k' . The SAT problem is built in two steps. First, we select the portion T of the background theory relevant to the contextualized interpretation $w(k)$ and $w(k')$, then we compute the logical relation between $w(k)$ and $w(k')$ which are implied by T .

Definition 4.3. *Let $\phi = w(k)$ and $\psi = w(k')$ be the contextualized interpretation of two concepts k and k' of two contexts c and c' , respectively. Let B be a theory (= logically closed set of axioms) in the logic where ϕ and ψ are expressed. The portion of B relevant to ϕ and ψ , is a subset T of B such that T contains all the axioms of B containing some concept occurring in ϕ or ψ .*

Clearly different contexts can be associated to different background theories, which encodes general and domain specific information. This information is stored in the context external assumptions under the field “domain”. Furthermore, when we determine the mapping between two contexts c_s and

WORDNET relation	Domain axiom
$t\#k =_w s\#h$	$t\#k \equiv s\#h$
$t\#k \leq_w s\#h$	$t\#k \sqsubseteq s\#h$
$t\#k \geq_w s\#h$	$t\#k \sqsupseteq s\#h$
$t\#k \perp_w s\#h$	$\neg t\#k \sqsubseteq s\#h$

Table 1: Encoding WORDNET relations in T-Box axioms

c_t we can take the perspective (i.e., the background theory) of the source or that of the target. The two perspectives indeed might not coincide. This justifies the introduction of directionality in the mapping. I.e. $c_s \xrightarrow{\subseteq} c_t$ means that c_s is more general than c_t according to the target perspective; while the relation $c_t \xrightarrow{\supseteq} c_s$ represent the fact that c_s is more general than c_t according to the source perspective.

In the first version of our matching algorithm we consider one a background theory B determined by transforming the WORDNET relations in a set of axioms in description logic, as shown in Table 1. In this table we introduce the notation \equiv_w , \leq_w , \geq_w , and \perp_w to represent the following relation between senses stored in WORDNET.

1. $s\#k \equiv_w t\#h$: $s\#k$ and $t\#h$ are synonyms (i.e., they are in the same synset);
2. $s\#k \leq_w t\#h$: $s\#k$ is either a hyponym or a meronym of $t\#h$;
3. $s\#k \geq_w t\#h$: $s\#k$ is either a hypernym or a holonym of $t\#h$;
4. $s\#k \perp_w t\#h$: $s\#k$ belongs to the set of opposite meanings of $t\#h$ (if $s\#k$ and $t\#h$ are adjectives) or, in case of nouns, that $s\#k$ and $t\#h$ are different hyponyms of the same synset.

In the extraction of the theory B from WORDNET we adopt a certain heuristic which turns out to perform satisfactory (see section on experimentation and evaluation). However, different sources as, specific domain ontologies, domain taxonomies, etc. and different heuristics can be used to build the theory B , from which T is extracted.

Going back to how we build the theory B , suppose, for example, that we want to discover the relation between *Chat and Forum* in the Google directory and *Chat and Forum* in the Yahoo directory in Figure 1. From WORDNET we can extract the following relevant axioms:

$$\text{art\#1} \sqsubseteq \text{humanities\#1}$$

(the sense 1 of ‘art’ is an hyponym of the sense 1 of ‘humanities’), and

$$\text{humanities\#1} \sqsupseteq \text{literature\#2}$$

(the sense 1 of ‘humanities’ is an hypernym of the sense 2 of ‘literature’).

The axioms extracted from WORDNET can now be used to check what mapping (if any) exists between k and k' looking at their contextualized interpretation. But which are the logical relations of $w(k)$ and $w(k')$ that encodes a mapping function between k and k' as given in Definition 3.3? Again,

relation	SAT Problem
$k_s \xrightarrow{\supseteq} k_t$	$T_t \models w(k_t) \sqsubseteq w(k_s)$
$k_s \xrightarrow{\subseteq} k_t$	$T_t \models w(k_s) \sqsubseteq w(k_t)$
$k_s \xrightarrow{\perp} k_t$	$T_t \models w(k_s) \sqcap w(k_t) \sqsubseteq \perp$
$k_s \xrightarrow{\equiv} k_t$	$T_t \models w(k_t) \sqsubseteq w(k_s) \text{ and } T_t \models w(k_s) \sqsubseteq w(k_t)$
$k_s \xrightarrow{*} k_t$	$w(k_s) \sqcap w(k_t) \text{ is consistent in } T_t$

Table 2: Verifying relations as a SAT problem

the encoding of the mapping into a logical relation is a matter of heuristics. Here we propose the translation described in Table 2. In this table T_t is the portion of the background theory of c_t relevant to k_s and k_t . The idea under this translation is to see WORDNET senses (contained in $w(k)$ and $w(k')$) as sets of documents. For instance the concept `art#1`, corresponding to the first WORDNET sense of `art`, is thought as the set of documents speaking about art in the first sense. Using the set theoretic interpretation of mapping given in definition 3.4, we have that mapping can be translated in terms of subsumption of $w(k)$ and $w(k')$. Indeed subsumption relation semantically corresponds to the subset relation.

So, the problem of checking whether *Chat and Forum* in Google is, say, less general than *Chat and Forum* in Yahoo amounts to a problem of satisfiability on the following formula:

$$\text{art\#1} \sqsubseteq \text{humanities\#1} \quad (1)$$

$$\text{humanities\#1} \sqsupseteq \text{literature\#2} \quad (2)$$

$$(\text{art\#1} \sqcap \text{literature\#2} \sqcap (\text{chat\#1} \sqcup \text{forum\#1})) \quad (3)$$

$$(\text{art\#1} \sqcup \text{humanities\#1}) \sqcap \text{humanities\#1} \sqcap (\text{chat\#1} \sqcup \text{forum\#1}) \quad (4)$$

It is easy to see that from the above axioms we can infer (3) \sqsubseteq (4).

To each relation it is possible to associate also a quantitative measure. For instance the relation “ c is compatible with d ” can be associated with a degree, representing the percentage of models that satisfy $\phi \sqcap \psi$ on the models that satisfy $\phi \sqcup \psi$. Another example is the measure that can be associated to the relation “ c is more general than d ” which is the percentage of the models of that satisfy ϕ on the models that satisfy ψ . This measure give a first estimation on how much ψ is a generalization of ϕ , the lower percentage, the higher generalization.

4.3 Implementation and evaluation

The algorithm has been implemented and tested as part of a peer-to-peer infrastructure for Distributed Knowledge Management. A detailed discussion of this aspect is described in another paper submitted to this conference. Here we summarize the main features and limitations of the current implementation, and the points that will be inserted in the future version. The result of the matching algorithm on the two contexts shown in Figure 1 are reported in the following table:

nodes of the source and target context	13, 14	
possible relations	182	100%
k_s is equivalent to k_t	4	2%
k_s is more general than k_t	24	13%
k_s is more specific than k_t	42	23%
k_s is compatible with k_t	98	54%
Total # or relations	168	92%

Some of the more interesting relations discovered by the algorithm are reported in Table 3.

5 Related work

Rahm and Bernstein [Rahm and Bernstein, 2001] suggest that there are three general strategies for matching schemas: *instance based* (using similarity between the objects (e.g., documents) associated to the schema to infer the relationship between the concepts); *schema-based* (determining the relationships between concepts analyzing the structure of a hierarchy and the meanings of the labels); and *hybrid* (a combination of the two strategies above). Our algorithm falls in the second group. In this section, we briefly compare our method with some of the most promising schema-based methods recently proposed, namely MOMIS [Bergamaschi *et al.*, 1999] a schema based semi automatic matcher, CUPID [Madhavan *et al.*, 2001; 2002] a schema based automatic matcher and GLUE [Doan *et al.*, 2002] an instance based automatic matcher.

The MOMIS (Mediator enviroNment for Multiple Information Sources) [Bergamaschi *et al.*, 1999] is a framework to perform information extraction and integration from both structured and semistructured data sources. It takes a global-as-view approach by defining a global integrated schema starting from a set of sources schema. In one of the first phases of the integration, MOMIS supports the discovery of overlapping (relations) between the different source schema. This is done by exploiting the knowledge in a Common Thesaurus with a combination of clustering techniques and Description Logics. The main differences between the matching algorithm implemented in MOMIS and CTXMATCH, is the fact that MOMIS, being an interactive process, which is a step of an integration procedure, does not support run-time generation of mappings.

More similar to CTXMATCH is the algorithm proposed in [Madhavan *et al.*, 2001], called CUPID. This is an algorithm for generic schema matching, based on a weighted combination of names, data types, constraints and structural matching. This algorithm exploits a limited amount of linguistic by associating a thesaurus to each schema, but differently from CTXMATCH does not uses the whole power of WORDNET. Another deeper differences between CUPID and CTXMATCH concerns the fact that CUPID can manage to discover the relation between to schemas S and T only when the S and the embedding of S in T are structurally isomorphic. CUPID seems not to deals in those cases were S and T are equivalent even if they have a completely different structure.

A different approach to ontology matching has been proposed in [Doan *et al.*, 2002]. Although the aim of the work (i.e. establishing mappings among concepts of overlapping ontologies) is in many respects similar to our goals,

Arts/Art history	≡	Arts & Humanities/Art History
Arts/Art history/Organizations	≡	Arts & Humanities/Art History/Organizations
Arts/Visual Art	⊆	Arts & Humanities/Visual Art
Arts/Visual Art/Photography	⊆	Arts & Humanities/Photography
Arts/Music/History/Baroque	⊆	Arts & Humanities/Art History
Arts/Art History	⊆	Arts & Humanities/Design Art/Architecture/History
Arts/Literature	⊆	Arts & Humanities/Humanities
Arts/Literature/Chat and forum	⊆	Arts & Humanities/Humanities/Chat and forum
Arts/Music/History/Baroque	≠	Arts & Humanities/Design Art/Architecture/History/Baroque

Table 3: Some results of CTXMATCH applied to the contexts in Figure 1

the methodologies differ significantly. A major difference is that the GLUE system builds mappings taking advantage of information contained in instances, while our current version of the CTXMATCH algorithm completely ignores them. This makes CTXMATCH more appealing, since most of the ontologies currently available on the Semantic Web still do not contain significant amount of instances. A second difference concerns the use of domain-dependent constraints, which, in case of the GLUE system, need to be provided manually by domain experts, while in CTXMATCH they are automatically extracted from an already existing resource (i.e. WordNet). Finally, CTXMATCH attempts to provide a qualitative characterization of the mapping in terms of the relation involved among two concepts, a feature which is not considered in GLUE. Although a strict comparison with the performances reported in [Doan *et al.*, 2002] is rather difficult, the accuracy achieved by CTXMATCH could be roughly compared with the accuracy of the GLUE module which uses less information (i.e. the “name learner”).

The problem of the integration and of the interoperability between different catalogs of overlapping domains is acquiring high relevance, not only in a commercial perspective (i.e. companies that want to exchange their products need to find mappings among their catalogs), but also on a scientific perspective [Schulten *et al.*, 2001; Agrawal and Srikant, 2001]

6 Conclusions

In the paper, we presented a first version of an algorithm for matching semantic schemas (viewed as contexts) via SAT. A lot of work remains to be done, and in particular: generalizing the types of structures we can match (beyond concept hierarchies); taking into account a larger collection of explicit assumptions; going beyond WORDNET as a source of linguistic and domain knowledge.

Finally, we are extensively testing the algorithm on large datasets made of real schemas. The preliminary results are described in a paper submitted to this conference. However, we observe that at the moment there is not a generally accepted methodology for comparing the results of different approaches to schema matching, and this makes it difficult to say which algorithm performs better in a given scenario, and to compare the results on the same examples.

References

- [Benerecetti *et al.*, 2000] M. Benerecetti, P. Bouquet, and C. Ghidini. Contextual Reasoning Distilled. *Journal of Theoretical and Experimental Artificial Intelligence*, 12(3):279–305, July 2000.
- [Bergamaschi *et al.*, 1999] Sonia Bergamaschi, Silvana Castano, and Maurizio Vincini. Semantic integration of semistructured and structured data sources. *SIGMOD Record*, 28(1):54–59, 1999.
- [Bonifacio *et al.*, 2002b] M. Bonifacio, P. Bouquet, and P. Traverso. Enabling distributed knowledge management. Managerial and technological implications. *Novatica and Informatik/Informatique*, III(1), 2002.
- [Borgida and Serafini, 2002] A. Borgida and L. Serafini. Distributed description logics: Directed domain correspondences in federated information sources. In R. Meersman and Z. Tari, editors, *On The Move to Meaningful Internet Systems 2002: CoopIS, Doa, and ODBase*, vol. 2519 of *LNCS*, pages 36–53. Springer Verlag, 2002.
- [Ghidini and Giunchiglia, 2001] C. Ghidini and F. Giunchiglia. Local models semantics, or contextual reasoning = locality + compatibility. *Artificial Intelligence*, 127(2):221–259, April 2001.
- [Madhavan *et al.*, 2001] Jayant Madhavan, Philip A. Bernstein, and Erhard Rahm. Generic schema matching with cupid. In *The VLDB Journal*, pages 49–58, 2001.
- [Madhavan *et al.*, 2002] Jayant Madhavan, Philip A. Bernstein, Pedro Domingos, and Alon Y. Halevy. Representing and reasoning about mappings between domain models. *Proc. of the 18th AAAI Canada*, pages 80–86, 2002.
- [Magnini *et al.*, 2002a] B. Magnini, L. Serafini, and M. Speranza. Linguistic based matching of local ontologies. In P. Bouquet, editor, *AAAI-02 WS on Meaning Negotiation*, Canada, 2002.
- [Rahm and Bernstein, 2001] Erhard Rahm and Philip A. Bernstein. A survey of approaches to automatic schema matching. *VLDB Journal*, 10(4):334–350, 2001.
- [Agrawal and Srikant, 2001] R. Agrawal and R. Srikant. On integrating catalogs. In *Proc. of the 10th Int. WWW Conference China*, 2001.
- [Doan *et al.*, 2002] A. Doan, J. Madhavan, P. Domingos, and A. Halevy. Learning to map between ontologies on the semantic web. In *Proc. of WWW-2002, 11th International WWW Conf.*, Hawaii 2002.
- [Schulten *et al.*, 2001] E. Schulten, et. al. Call for participants: The e-commerce product classification challenge. *IEEE Intelligent Systems*, 16(4), 2001.