Managing knowledge across distributed semantic peers

Matteo Bonifacio [1] and **Paolo Bouquet** [2] and **Luciano Serafini**[3] and **Stefano Zanobini**[4]

Abstract. The problem of finding an agreement on the meaning of heterogeneous knowledge sources is one of the key issues in the development of the distributed knowledge management applications. In this paper, we propose a new algorithm for discovering semantic mappings across heterogeneous schemas. This approach shifts the problem of semantic coordination from the problem of computing linguistic or structural similarities (what most other proposed approaches do) to the problem of deducing relations between sets of logical formulas that represent the meaning of nodes belonging to different schemas. We show how to apply the approach and the algorithm to an interesting family of schemas, namely hierarchical classifications. Finally, we argue why this is a significant improvement on previous approaches.

1 INTRODUCTION

Is well known how one of the major challenges that organizations are increasingly facing is to develop a capacity to manage knowledge assets [17]. Such challenge, although agreeable in principle, can be addressed in fundamentally different ways depending on how the very nature of knowledge is described. As shown elsewhere in more detail (see [8]), assuming knowledge as a 'content' that can be standardized in order to be shared, KM system will be configured as centralized repositories in which knowledge objects are semantically organized around a shared conceptualization (e.g an ontology or a taxonomy). On the other hand, assuming knowledge as an intrinsically contextual matter whose value lies in its relation with local perspectives and practices, KM systems will tend to be configured as constellations of autonomous knowledges. Here, each knowledge is organized around a local and often private conceptualization, and, as a consequence, knowledge exchange is achieved by means of semantic coordination rather than standardization. That is, different knowledge holders (such as informal groups or communities of practice [29]) need to establish communication rules and protocols able to create semantic bridges across heterogeneous semantic configurations ([5]).

In favour of the second approach, here named Distributed Knowledge Management [7][5], there are of course theoretical arguments; in particular, in the domains of organization sciences [28], social sciences ([12, 13, 21]) and cognitive studies of learning [3], an increasingly extended school of thought underlines how distributedness and diversity is a source of value, innovation, flexibility and adaptability (For a review see [10]. But here we want to underline a more practical issue. In fact, it is evident to practitioners how knowledge is increasingly spread across a wide set of technologies and applications and, moreover, how such diversity resembles different working practices. As a matter of fact, although organizational intelligence pursues the dream of collecting all the relevant knowledge within a single homogeneous source, technological and semantic heterogeneity seems to move in the opposite direction. Instead of being reduced, it seems rather to explode.

One of the key challenges in the development of open distributed systems is the attempt of enabling the exchange of meaningful information across applications which (i) may use autonomously developed schemas for organizing locally available data (a local context), and (ii) need to discover relations between schemas (or contexts) to achieve their users' goals. Typical examples are databases using different schemas, and document repositories using different classification structures.

In restricted environments, like a small corporate Intranet, this problem is typically addressed by introducing shared models (e.g., ontologies) throughout the entire organization[6]. The idea is that, once local schemas are mapped onto a shared ontology, the required relations between them is completely defined. However, in open environments (like the Web, large and complex firms, networked companies and networks of companies), for the theoretical reasons we said before, and for technical reasons, including the difficulty of 'negotiating' a shared model of data that suits the needs of all parties involved, and the practical impossibility of maintaining such a shared model in a highly dynamic environment, such an approach can't work. In this kind of scenarios, a more dynamic and flexible method is needed, where no shared model can be assumed to exist, and semantic relations between concepts belonging to different schemas must be discovered on-the-fly. In other words, we need a sort of peer-to-peer form of semantic coordination, in which two or more *semantic peers* (i.e., agents with autonomous data schemas) discover relations across their schemas and use them to provide the required services.

In this paper, we propose a very general approach to the problem of coordinating schemas of two or more semantic peers. The method is based on the idea that the mappings across local schemas we are interested in are semantic relations, namely must have a model-theoretic interpretation. This allows us to use standard theorem proving techniques to infer mappings across schemas, and to validate the results. We'll stress that this is extremely different from what other methods for discovering mappings across heterogeneous models do, as they are mostly based on a notion of similarity which is not – strictly speaking – semantic.

The method is then demonstrated on a significant instance of the problem, namely the problem of coordinating hierarchical classifications (HCs). HCs are tree–structures used for organizing/classifying data (such as documents, goods, activities, services). Some well-known examples of HCs are web directories (see e.g. the Google^{TM}

[1] University of Trento, Italy, Email: bonifacio@economia.unitn.it
[2] University of Trento, Italy, Email: bouquet@dit.unitn.it
[3] IRST, Trento, Italy – Email: serafini@itc.it
[4] University of Trento, Italy, Email: zanobini@dit.unitn.it

[5] But see also [1, 2] for other approaches in Distributed Knowledge Management.

[6] But see [7] for a discussion of the drawbacks of this approach from the standpoint of Knowledge Management applications.

Directory or the Yahoo!^{TM}Directory), file systems and document databases in content/knowledge management systems. The main technical contribution of this part is an algorithm, called CTX-MATCH, which takes as input two HCs H and H' and, for each pair of concepts $k \in H$ and $k' \in H'$, returns their semantic relation r (the whole set of triples $\langle k, k', r \rangle$ is called 'mapping'). The idea is that mappings across semantic models can then be used by other application to answer queries (e.g., by finding documents classified under an unknown category in another HC) or more in general to provide services which require an agreement on the meaning of terms.

With respect to other methods proposed in the literature (often under different "headings", such as schema matching, ontology mapping, semantic integration), the main innovation of our approach is that mappings across concepts belonging to different models are deduced via logical reasoning, rather then derived through some more or less complex heuristic procedure, and thus can be assigned a clearly defined model-theoretic semantics. This shifts the problem of coordinating semantic peers from the problem of computing linguistic or structural similarities (possibly with the help of a thesaurus and of other information about the type of arcs between nodes), to the problem of deducing relations between formulas that represent the meaning of each concept in a model. This explains, for example, why our approach performs much better than other ones when two concepts are intuitively equivalent, but occur in structurally very different HCs.

The paper goes as follows. In Section 2 we introduce the main conceptual assumptions of the new approach we propose to semantic coordination. In Section 3, we present the main features of CTX-MATCH, the proposed algorithm for coordinating HCs. In the final part of the paper, we sum-up the results of testing the algorithm on web directories and catalogs (Section 4) and compare our approach with other proposed approaches for matching schemas (Section 5).

2 OUR APPROACH

The method we propose assumes that we deal with a network of *semantic peers*, namely physically connected entities which can autonomously decide how to organize locally available data (in a sense, are semantically autonomous agents). Each peer organizes its data using one or more abstract schemas (e.g., database schemas, directories in a file system, classification schemas, taxonomies, and so on). Different peers may use different schemas to organize the same data collection, and conversely the same schemas can be used to organize different data collections.

We also assume that semantic peers need to exchange data (e.g. documents classified under different categories in their local classification schemas) to execute complex tasks. To do this, each semantic peer needs to compute mappings between its local schema and other peers' schemas. Intuitively, a mapping can be viewed as a set of pairwise relations between elements of two distinct schemas.

The first idea behind our approach is that mappings must be semantic relations, namely relations with a well-defined model-theoretic interpretation. This is an important difference with respect to approaches based on matching techniques, where a mapping is a measure of (linguistic, structural, ...) similarity between schemas (e.g., a real number between 0 and 1). The main problem with the latter techniques is that the interpretation of their results is an open problem. For example, how should we interpret a 0.9 similarity? Does it mean that one concept is slightly more general than the other one? Or maybe slightly less general? Or that their meaning 90% overlaps (whatever that means)? Instead, our method returns semantic relations, e.g. that the two concepts are (logically) equivalent, or that one is (logically) more/less general, or that they are mutually exclusive. As we will argue, this gives us many advantages, essentially related to the consequences we can infer from the discovery of such a relation[7].

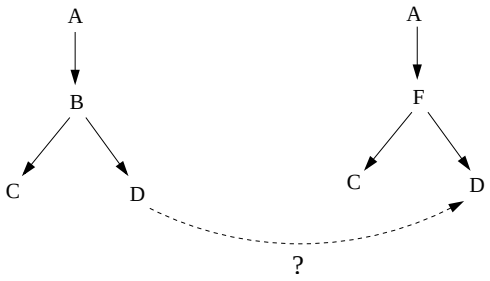

Figure 1. Mapping abstract structures

The second idea is that, to discover semantic relations, one must make explicit the meaning implicit in each element of a schema. The claim is that making explicit the meaning of elements in a schema is the only way of computing semantic relations between elements of distinct schemas, and that this can be done only for schemas in which meaningful labels are used. If this is true, then addressing the problem of discovering semantic relations as a problem of matching abstract graphs is conceptually wrong. To illustrate this point, consider the difference between the problem of mapping abstract schemas (like those in Figure 1) and the problem of mapping schemas with meaningful labels (like those in Figure 2). Nodes in abstract schemas do not have an implicit meaning, and therefore, whatever technique we use to map them, we will find that there is some relation between the two nodes `D` in the two schemas which depends only on the abstract form of the two schemas. The situation is completely different for schemas with meaningful labels, as we can make explicit a lot of information that we have about the terms which appear in the graph, and their relations (e.g., that Tuscany is part of Italy, that Florence is in Tuscany, and so on). It's only this information which allows us to understand why the semantic relation between the two nodes `MOUNTAIN` and the two nodes `FLORENCE` is different, despite the fact that the two pairs of schemas are structurally equivalent between them, and both are structurally isomorphic with the pair of abstract schemas in Figure 1. Indeed, for the first pair of nodes, the set of documents we would classify under the node `MOUNTAIN` on the left hand side is a subset of the documents we would classify under the node `MOUNTAIN` on the right; whereas the set of documents which we would classify under the node `FLORENCE` in the left schema is exactly the same as the set of documents we would classify under the node `FLORENCE` on the right hand side.

As a consequence, our method is mainly applied to schemas with labels which are meaningful for the community of their users. This gives us the chance of exploiting the complex degree of semantic coordination implicit in the way a community uses the language from which the labels are taken. Notice that the status of this linguistic coordination at a given time is already 'codified' in artifacts (e.g., dictionaries, but today also ontologies and other formalized models), which provide senses for words and more complex

[7] For a more detailed discussion of the distinction between syntactic and semantic methods, see [19].

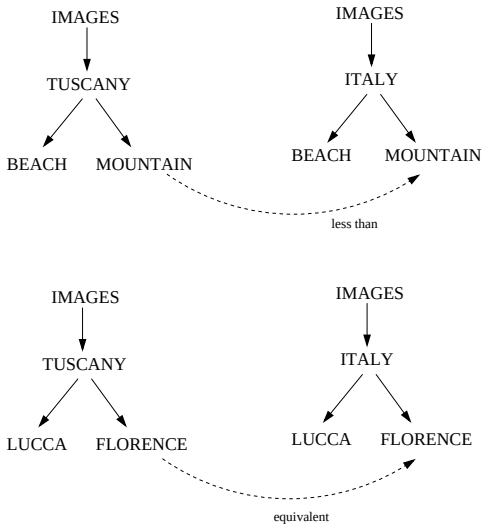

Figure 2. Mapping schemas with meaningful labels

expressions, relations between senses, and other important knowledge about them. Our aim is to exploit these artifacts as an essential source of constraints on possible/acceptable mappings across structures. The method is based on the explicitation of the meaning associated to each node in a schema (notice that schemas such as the two classifications in Figure 2 are not semantic models themselves, as they do not have the purpose of defining the meaning of terms they contain; however, they presuppose a semantic model, and indeed that's the only reason why we humans can read them quite easily). The explicitation process uses three different levels of knowledge:

Lexical knowledge : It is the knowledge about the words used in the labels. For example, the fact that the word 'Florence' can be used to indicate 'a city in Italy' or 'a city in the South Carolina' (homonymy), and to handle the synonymy;

Knowledge Base : It is the knowledge about the relation between the concepts expressed by words. For example, the fact that Tuscany is part of Italy, or that Florence is in Italy;

Structural knowledge : It is the knowledge deriving from how labeled nodes are arranged in a given schema. For example, the fact that the node labeled `MOUNTAIN` is below a node `IMAGES` tells us that it classifies images of mountains, and not, say, books about mountains.

As an example of how the three levels are used, consider again the mapping between the two nodes `MOUNTAIN` of Figure 2. Lexical knowledge is used to determine what concepts can be expressed by each label, e.g. that the word 'Images' can denote the concept 'a visual representation produced on a surface'. Knowledge Base tells us, among other things, that Tuscany is part of Italy. Finally, structural knowledge tells us that the intended meanings of the two nodes `MOUNTAIN` is 'images of Tuscan mountains' on the left hand side, and 'images of Italian mountains' on the right hand side. Using this information, human reasoners (i) understand the meaning expressed by the left hand node, ('images of Tuscan mountains', denoted by P), (ii) understand the meaning expressed by the right hand node ('images of Italian mountains', denoted by P'), and finally (iii) understand the semantic relation between the meaning of the two nodes, namely that $P \subseteq P'$.

These three levels of knowledge are used to produce a new, richer representation of the schema, where the meaning of each node is made explicit and encoded as a logical formula and a set of axioms. This formula is an approximation of the meaning of the node when it occurs in that schema. The problem of discovering the semantic relation between two nodes can now be stated not as a matching problem, but as a relatively simple problem of logical deduction. Intuitively, as we will say in a more technical form in the rest of the paper, determining whether there is an equivalence relation between the meaning of two nodes can be encoded as a problem of testing whether the first implies the second and vice versa (given a suitable collection of axioms, which acts as a sort of background theory); and determining whether one is less general than the other one amounts to testing if the first implies the second. As we will say, in the current version of the algorithm we encode this reasoning problem as a problem of logical satisfiability, and then compute mappings by feeding the problem to a standard SAT solver.

3 THE ALGORITHM: CTXMATCH

In this section we show how to apply the general approach described in the previous section to the problem of coordinating *Hierarchical Classifications* (hereafter HCs), namely concept hierarchies [14] used for grouping documents in categories.

In our approach, we assume the presence of a network of semantic peers, where each peer is defined as follows:

Definition 1 *A semantic Peer is a triple $\langle \mathcal{D}, \mathcal{S}, \langle L, O \rangle \rangle$, where:*

- *$\mathcal{D}$ is a set of documents;*
- *$\mathcal{S}$ represents the set of schemas used by the peer for organizing its data;*
- *$\langle L, O \rangle$ is a pair composed by a lexicon L and a knowledge base representation O.*

The structure of the semantic peer reflects the three levels of knowledge we showed before: $\mathcal{S}$ represents structural knowledge, L contains lexical knowledge, and O is knowledge base. Formally, L is a repository of pairs $\langle w, C \rangle$, where w is a word and C is a set of concepts. Each pair $\langle w, C \rangle$ represents the set of concepts C denoted by a word w. For example, a possible entry for a lexicon should express that the word 'fish' can denote at least two concepts: 'an aquatic vertebrate' and 'the twelfth sign of zodiac'. An important example of this kind of repository is represented by WORDNET [20]. A knowledge base O expresses the set of relations holding between different concepts. For example, a knowledge base O should express that the concept 'an aquatic vertebrate' denoted by the word 'fish' stays in a *IsA* relation with the concept of 'animal' ('fish are animals') and that the concept 'the twelfth sign of zodiac' denoted by the same word 'fish' stays in a *IsA* relations with a geometrical shape ('fish is a geometrical shape'). Formally, knowledge base is a logical theory written is a specific language, as for example Prolog clauses, RDF triples, DAML/OIL, OWL.

Our method is designed for the following scenario: a peer A (called the *seeker*) needs to find new documents relative to some category in one of its HCs, S. Imagine that peer A knew that peer B (the provider) owns interesting documents, and imagine that B classify its documents by means of a HC S'. The problem of finding such documents can be solved in a standard way: discovering a mapping between the two structures. Formally, a mapping can be defined as:

Definition 2 *A mapping $\mathcal{M}$ between two schemas S and S' is a set of triples $\{\langle m, n, R\rangle \mid m \in S, n \in S'\}$, where R is a semantic relation between m and n.*

In this version of the algorithm, five relations are allowed between nodes of different HCs: $m \supseteq n$ (m is more general than n); $m \subseteq n$ (m is less general than n); $m \equiv n$ (m is equivalent to n); $m \cap n$ (m is compatible with n); $m \perp n$ (m is disjoint from n).

The algorithm CTXMATCH takes as **inputs** the HC S of the seeker and the HC S', the lexicon L and the knowledge base O of the provider[8]. As we will show in the following, the lexicon L and the knowledge base O play a major part in determining the mapping between schemas. But, from the definition of semantic peer follows that each peer has its own lexicon and knowledge base. A consequence of this consideration is that the mapping returned by the algorithm expresses the point of view (regarding the mapping) of the provider, and, consequently, is directional: the seeker, *mutata mutandis*, can find a different mapping. The **output** of the algorithm will be a mapping $\mathcal{M}$.

Algorithm 1 CTXMATCH(S,$S'L, O$)
▷ *Hierarchical classifications S, S'*
▷ *Lexicon L*
▷ *knowledge base O*
VarDeclarations
contextualized concept $\langle \phi, \Theta\rangle, \langle \psi, \Upsilon\rangle$
relation R
mapping $\mathcal{M}$
1 **for** each pair of nodes m, n, $m \in S$ and $n \in S'$ **do**
2 $\langle \phi, \Theta\rangle \leftarrow$ SEMANTIC–EXPLICITATION(m, S, L, O);
3 $\langle \psi, \Upsilon\rangle \leftarrow$ SEMANTIC–EXPLICITATION(n, S', L, O);
4 $R \leftarrow$ SEMANTIC–COMPARISON$(\langle \phi, \Theta\rangle, \langle \psi, \Upsilon\rangle, O)$;
5 $\mathcal{M} \leftarrow \mathcal{M} \cup \langle m, n, R\rangle$;
6 **Return** M;

The algorithm has essentially the following two main macro steps.

Steps 2–3 : in this phase, called *Semantic explicitation*, the algorithm tries to interpret pair of nodes m, n in the respective HCs S and S' by means of the lexicon L and the knowledge base O. The idea is trying to generate a formula approximating the meaning expressed by a node in a structure (ϕ), and a set of axioms formalizing the suitable knowledge base (Θ). Consider, for example, the node FLORENCE in left lower HC of Figure 2: steps 2–3 will generate a formula approximating the statement 'Images of Florence in Tuscany' (ϕ) and an axiom approximating the statement 'Florence is in Tuscany' (Θ). The pair $\langle \phi, \Theta\rangle$, called *contextualized concept*, expresses, in our opinion, the meaning of a node in a structure.

Step 4 : in this phase, called *Semantic comparison*, the problem of finding the semantic relation between two nodes m and n is encoded as the problem of finding the semantic relation holding between two contextualized concepts, $\langle \phi, \Theta\rangle$ and $\langle \psi, \Upsilon\rangle$.

Finally, step 5 generates the mapping simply by reiteration of the same process over all the possible pair of nodes $m \in S$ $n \in S'$ and step 6 returns the mapping.

The two following sections describe in detail these two top-level operations, implemented by the functions SEMANTIC–EXPLICITATION and SEMANTIC–COMPARISON.

[8] In the version of the algorithm presented here, we use WORDNET as a source of both lexical and knowledge base. However, WORDNET could be replaced by another combination of a linguistic resource and a knowledge base resource.

3.1 SEMANTIC EXPLICITATION

In this phase we make explicit in a logical formula[9] the meaning of a node into a structure, by means of a lexical and a knowledge base. In steps 1 and 2, the function EXTRACT–CANDIDATE–CONCEPTS uses lexical knowledge to associate to each word occurring in the nodes of an HC all the possible concepts denoted by the word itself. Consider the lower left structure of Figure 2. The label 'Florence' is associated with two concepts, provided by the lexicon (WORDNET), corresponding to 'a city in central Italy on the Arno' (`florence#1`) or a 'a town in northeast South Carolina' (`florence#2`). In order to maximize the possibility of finding an entry into the Lexicon, we use both a postagger and a lemmatizator over the labels.

Algorithm 2 SEMANTIC–EXPLICITATION(t, S, L, O)
▷ *t is a node in S*
▷ *structure S*
▷ *lexicon L*
▷ *knowledge base O*
VarDeclarations
single concept $con[]$
set of formulas Σ
formula δ
1 **for** each node n in S **do**
2 $con[n] \leftarrow$ EXTRACT–CANDIDATE–CONCEPTS(n, L);
3 $\Sigma \leftarrow$ EXTRACT–LOCAL-AXIOMS$(t, S, con[], O)$;
4 $con[] \leftarrow$ FILTER–CONCEPTS$(S, \Sigma, con[])$;
5 $\delta \leftarrow$ BUILD–COMPLEX–CONCEPT$(t, S, con[])$;
6 **Return** $\langle \delta, \Sigma\rangle$;

In the step 3, the function EXTRACT–LOCAL–AXIOMS tries to define the ontological relations existing between the concepts in a structure. Consider again the left lower structure of Figure 2. Imagine that the concept 'a region in central Italy' (`tuscany#1`) has been associated to the node TUSCANY. The function EXTRACT–LOCAL–AXIOMS has the aim to discover if it exists some kind of relation between the concepts `tuscany#1`, `florence#1` and `florence#2` (associated to node FLORENCE). Exploiting knowledge base resource we can discover, for example, that '`florence#1` PartOf `tuscany#1`', i.e. that exists a 'part of' relation between the first sense of 'Florence' and the first sense of Tuscany.

Knowledge base relations are translated into logical axioms, according to Table 1. So, the relation '`florence#1` PartOf `tuscany#1`' is encoded as '`florence#1` $\rightarrow$ `tuscany#1`'[10].

WORDNET relation	axiom
s#k synonym t#h	s#k $\equiv$ t#h
s#k { hyponym | PartOf }t#h	s#k $\rightarrow$ t#h
s#k { hypernym | HasPart }t#h	t#h $\rightarrow$ s#k
s#k contradiction t#h	$\neg$(t#k $\wedge$ s#h)

Table 1. WORDNET relations and their axioms.

Step 4 has the goal of filtering out unlikely senses associated to each node. Going back to the previous example, we try to discard one of the senses associated to node FLORENCE. Intuitively,

[9] The choice of the logics depends on how expressive one wants to be in the approximation of the meaning of nodes, and on the complexity of the NLP techniques used to process labels. In our first implementation we adopted propositional logic, where each propositional letter corresponds to a concept (synset) provided by WORDNET.

[10] For heuristical reasons – see [11] – we consider only relations between concepts on the same path of a HC and their siblings.

the sense 2 of 'Florence', as 'a town in northeast South Carolina' (florence#2), can be discarded, because the node FLORENCE refers clearly to the city in Tuscany. We reach this result by analyzing the extracted local axioms: the presence of an axiom such as 'florence#1 PartOf tuscany#1' is used to make the conjecture that the contextually relevant sense of Florence is the city in Tuscany, and not the city in USA. When ambiguity persists (axioms related to different senses or no axioms at all), all the possible senses are left and encoded as a disjunction.

Step 5 has the objective of building a complex concept (i.e., the meaning of a node label when it occurs in a specific position in a schema) for nodes in HCs. As described in [11], node labels are *singularly* processed by means of NLP techniques and translated into a logical formula[11]. The result of this first process is that each node has a preliminary interpretation, called *simple concept*, which doesn't consider the position of the node in the structure. For example, the simple concept associated to the node FLORENCE of the lower left hand structure of Figure 2 is trivially the atom florence#1 (i.e. one of the two senses provided by WORDNET and not discarded by the filtering). Then, these results are combined for generating a formula approximating the meaning expressed by a node *into a structure*. In this version of the algorithm, we choose to express the meaning of a node n as the conjunction of the simple concepts associated to the nodes lying in the path from root to n. So, the formula approximating the meaning expressed by the same node FLORENCE *into the HC* is $(\texttt{image\#1} \vee \ldots \vee \texttt{image\#8}) \wedge \texttt{tuscany\#1} \wedge \texttt{florence\#1}$.

Step 6 returns the formula expressing the meaning of the node and the set of local axioms founded by step 3.

3.2 SEMANTIC COMPARISON

This phase has the goal of finding the semantic relation holding between two contextualized concepts (associated to two nodes in different HCs).

Algorithm 3 SEM–COMP($\langle \phi, \Theta \rangle$,$\langle \psi, \Upsilon \rangle$,$O$)
▷ *contextualized concept* $\langle \phi, \Theta \rangle$
▷ *contextualized concept* $\langle \psi, \Upsilon \rangle$
▷ *knowledge base* O

VarDeclarations
set of formulas Γ
relation R

1 $\Gamma \leftarrow$ EXTRACT–RELATIONAL–AXIOMS(ϕ, ψ, O);
2 $R \leftarrow$ FIND–SEM–REL($\langle \phi, \Theta \rangle$, $\langle \psi, \Upsilon \rangle$, Γ)
3 **Return** R;

In Step 1, the function EXTRACT–RELATIONAL–AXIOMS tries to find axioms which connect concepts belonging to different HCs. The process is the same as that of function EXTRACT–LOCAL–AXIOMS, described above. Consider, for example, the senses italy#1 and tuscany#1 associated respectively to nodes ITALY and TUSCANY of Figure 2: the relational axioms express the fact that, for example, 'Tuscany PartOf Italy' ($\texttt{tuscany\#1} \rightarrow \texttt{italy\#1}$).

The problem of finding the semantic relation between two nodes n and m (line 2) is encoded into a satisfiability problem involving both the contextualized concepts associated to the nodes and the relational axioms extracted in the previous phases. In particular, the function FIND-SEM-REL is defined in the algorithm 4.

[11] Although in this paper we present very simple examples, the NLP techniques exploited in this phase allow us to handle labels containing complex expressions, as conjunctions, commas, prepositions, expressions denoting exclusion, like 'except' or 'but not', multiwords and so on.

Algorithm 4 FIND–SEM–REL($\langle \phi, \Theta \rangle$, $\langle \psi, \Delta \rangle$, Γ)
▷ *contextualized concept* $\langle \phi, \Theta \rangle$, $\langle \psi, \Delta \rangle$
▷ *set of formulas* Γ

VarDeclarations
semantic relation R

1 **if** $\Theta, \Delta, \Gamma \models \neg(\phi \wedge \psi)$ **then** $R \leftarrow \perp$;
2 **else if** $\Theta, \Delta, \Gamma \models (\phi \equiv \psi)$ **then** $R \leftarrow \equiv$;
3 **else if** $\Theta, \Delta, \Gamma \models (\phi \rightarrow \psi)$ **then** $R \leftarrow \subseteq$;
4 **else if** $\Theta, \Delta, \Gamma \models (\psi \rightarrow \phi)$ **then** $R \leftarrow \supseteq$;
5 **else** $R \leftarrow \cap$;
6 **Return** R;

So, to prove whether the two nodes labeled FLORENCE in Figure 2 are equivalent, we check the logical equivalence between the formulas approximating the meaning of the two nodes, given the local and the relational axioms. Formally, we have the following satisfiability problem:

Θ	$\texttt{florence\#1} \rightarrow \texttt{tuscany\#1}$
ϕ	$(\texttt{image\#1} \vee \ldots \vee \texttt{image\#8}) \wedge \texttt{tuscany\#1} \wedge \texttt{florence\#1}$
Δ	$\texttt{florence\#1} \rightarrow \texttt{italy\#1}$
ψ	$(\texttt{image\#1} \vee \ldots \vee \texttt{image\#8}) \wedge \texttt{italy\#1} \wedge \texttt{florence\#1}$
Γ	$\texttt{tuscany\#1} \rightarrow \texttt{italy\#1}$

It is simple to see that the returned relation is '$\equiv$'. Note that the satisfiability problem for finding the semantic relation between the nodes MOUNTAIN of Figure 2 is the following:

Θ	$\emptyset$
ϕ	$(\texttt{image\#1} \vee \ldots \vee \texttt{image\#8}) \wedge \texttt{tuscany\#1} \wedge \texttt{mountain\#1}$
Δ	$\emptyset$
ψ	$(\texttt{image\#1} \vee \ldots \vee \texttt{image\#8}) \wedge \texttt{italy\#1} \wedge \texttt{mountain\#1}$
Γ	$\texttt{tuscany\#1} \rightarrow \texttt{italy\#1}$

The returned relation is '$\subseteq$'.

4 TESTING THE ALGORITHM

In this section, we report from [23] some results of the first test on CTXMATCH on real HCs (i.e., pre-existing classifications used in real applications).

4.1 USE CASE PRODUCT RE-CLASSIFICATION

In order to centrally manage all the company acquisition processes, the headquarter of a well known worldwide telecommunication company had realized an e-procurement system[12], which all the company branch-quarters were required to join. Each single office was also required to migrate from the product catalogue they used to manage, to this new one managed within the platform. This catalogue is extracted from the Universal Standard Products and Services Classification (UNSPSC), which is an open global coding system that classifies products and services. The UNSPSC is used extensively around the world in the electronic catalogues, search engines, procurement application systems and accounting systems. UNSPSC is a four-level hierarchical classification; an extract is reported in the following table:

Level 1 **Furniture and Furnishings**
Level 2 **Accommodation furniture**
Level 3 **Furniture**
Level 4 **Stands**
Level 4 **Sofas**
Level 4 **Coat racks**

[12] An e-procurement system is a technological platform which supports a company in managing its procurement processes and, more in general, the re-organization of the value chain on the supply side.

The Italian office asked us to apply the matching algorithm to re-classify into UNSPSC (version 5.0.2) the catalogue of office equipment and accessories used to classify company suppliers. The result of running CTXMATCH over UNSPSC and the catalogue can be cleary interpreted in terms of re-classification: if the algorithm returns that the item i of the catalogue is equivalent to, or more specific than, the node c_{UNSPSC} of UNSPSC, then i can be classified under c_{UNSPSC} of UNSPSC.

The items to be re-classified are mainly labeled with Italian phrases, but labels also contain abbreviations, acronyms, proper names, some English phrases and some typing errors. The English translation of an extract of this list is reported in the following table. The italic parts were contained in the original labels.

Code	Description
ENT.21.13	cartridge *hp desk jet 2000c*
ENR.00.20	magnetic tape cassette *exatape 160m xl 7,0gb*
ESA.11.52	*hybrid roller pentel* red
EVM.00.40	safety scissors, length 25 cm

The item list was matched with two UNSPSC's-segments, namely: *Office Equipment and Accessories and Supplies* (segment 44) and *Paper Materials and Products* (segment 14).

Notice that the company item catalogue we had to deal with was a plain list of items, each identified with a numerical code composed of two numbers, the first referring to a set of more general categories. For example, the number 21 at the beginning of *21.13-cartridge hp desk jet 2000c* corresponds to *printer tapes, cartridge and toner*. We first normalized and matched the plain list against UNSPSC. This did not lead us to a satisfactory result. The algorithm performed much better when we made the hierarchical classification contained in the item codes explicit. This was done by substituting the first numerical code of each item with their textual description provided by experts of the company.

After running CTXMATCH, the validation phase of our results was made by comparing them with the results of a simple keyword-based algorithm. Obviously, in order to establish the correctness of results in terms of precision and recall we have to compare them with a correct and complete matching list. Not having such a list, we asked a domain expert, Alessandro Cederle, Managing Director of Kompass Italia[13] to manually validate them.

4.2 RESULTS

This section presents the results of the re-classification phase. Consider first the baseline matching process. The baseline was performed by a simple keyword-based matching that worked according to the following rule:

> *for each item description (made up of one or more words) return the set of nodes, and their paths, which maximize the occurrences of the item words*

The following tables summarizes the results of the baseline matching:

	Baseline classification	
Total items	194	100%
Rightly classified	75	39%
Wrongly classified	92	47%
Non classified	27	14%

Given the 194 items to be re-classified, the baseline process found 1945 possible nodes in UNSPSC. This means that for each item the baseline found an average of 6 possible classifications. What is crucial is that only 75 out of the 1945 proposed nodes are correct. The baseline, being a simple string matching, is able to capture a certain number of re-classifications. However the percentage of error is quite high (47%) with respect to the one of correctness (39%). The results of the matching algorithm are reported in the following table:

	CTXMATCH **classification**	
Total items	194	100%
Rightly classified	136	70%
Wrongly classified	16	8%
Non classified	42	22%

In this case, the percentage of success is sensibly higher (70%) and, even more relevant, the percentage of error is minimal (8%)[14]. This is also confirmed by the values of precision and recall, computed with respect to the validated list:

	Total matches	**Precision**	**Recall**
Baseline	1945	4%	39%
CTXMATCH	641	21%	70%

The baseline precision level is quite small, while the matching one is not excellent, but definitely better. The same observations can be made also for the recall values.

If there are not enough information to infer semantic relation, CTXMATCH returns a percentage, which is intended to represent the degree of compatibility between the two elements. Degree of compatibility is computed on the basis of a linguistic co-occurrence measures.

As for as the Non Classified items, notice that:

- In some cases, the item to be re-classified were not correctly classified in the company catalogue. Therefore, CTXMATCH could not compute the relations with the node and its father node, in the right way. Examples are: *ashtray* was classified under *tape dispenser*; *wrapping paper* was classified under *adhesive labels*.
- In other cases, semantic coordination was not discovered due to a lack of knowledge base. For instance to match *paper for hp* with UNSPSC class of printer paper it would have been necessary to know that *hp* stands for Helwett Packard, and that it is a company which produces printers.

In a further experiment, we run CTXMATCH between the company catalogue (in italian) and the English version of UNSPSC. This was possible because the matching is computed on the basis of the WORDNET sense IDs, and in the version of WORDNET we used, wordnet-senses ID of italian and english words are aligned (i.e., the wordnet-sense ID associated to word and its translation in the other

[13] Kompass (www.kompass.com) is a company which provides product information, contacts and other information about 1.8 million companies worldwide. All companies are classifi ed under the Kompass Product Classifi cation with more than 52,000 products and services.

[14] Notice that the algorithm did not take into account just the UNSPSC level 4 category, since in some cases catalogues items can be matched with UNSPSC level 3 category nodes.

language is the same). This experiment allows us to find more semantic matches.

More in general, this way allows us to approach and manage multilanguage environments and to exploit the richness which typically characterizes the English version of linguistic resources[15].

5 RELATED WORK

CTXMATCH shifts the problem of semantic coordination from the problem of matching (in a more or less sophisticated way) semantic structures (e.g., schemas) to the problem of deducing semantic relations between sets of logical formulae. Under this respect, to the best of our knowledge, there are no other works to which we can compare ours.

However, it is important to see how CTXMATCH compares with the performance of techniques based on different approaches to semantic coordination. There are four other families of approaches that we will consider: graph matching, automatic schema matching, semi-automatic schema matching, and instance based matching. For each of them, we will discuss the proposal that, in our opinion, is more significant. The comparison is based on the following five dimensions: (1) if and how structural knowledge is used; (2) if and how lexical knowledge is used; (3) if and how knowledge base is used; (4) if instances are considered; (5) the type of result returned. The general results of our comparison are reported in Table 2.

In graph matching techniques, a concept hierarchy is viewed as a tree of labelled nodes, but the semantic information associated to labels is substantially ignored. In this approach, matching two graphs G_1 and G_2 means finding a sub-graph of G_2 which is isomorphic to G_2 and report as a result the mapping of nodes of G_1 into the nodes of G_2. These approaches consider only structural knowledge and completely ignore lexical knowledge and knowledge base. Some examples of this approach are described in [30, 27, 26, 24, 16].

CUPID [22] is a completely automatic algorithm for schema matching. Lexical knowledge is exploited for discovering linguistic similarity between labels (e.g., using synonyms), while the schema structure is used as a matching constraint. That is, the more the structure of the subtree of a node s is similar to the structure of a subtree of a node t, the more s is similar to t. For this reason CUPID is more effective in matching concept hierarchies that represent data types rather than hierarchical classifications. With hierarchical classifications, there are cases of equivalent concepts occurring in completely different structures, and completely independent concepts that belong to isomorphic structures. Two simple examples are depicted in Figure 3. In case (a), CUPID does not match the two nodes labelled with `ITALY`; in case (b) CUPID finds a match between the node labelled with `FRANCE` and `ENGLAND`. The reason is that CUPID combines in an additive way lexical and structural information, so when structural similarity is very strong (for example, all neighbor nodes do match), then a relation between nodes is inferred without considering labels. So, for example, `FRANCE` and `ENGLAND` match because the structural similarity of the neighbor nodes is so strong that labels are ignored.

MOMIS (Mediator envirOnment for Multiple Information Sources) [4] is a set of tools for information integration of (semi-)structured data sources, whose main objective is to define a global schema that allows a uniform and transparent access to the data stored in a set of semantically heterogeneous sources. One of the key steps of MOMIS is the discovery of overlappings (relations) between the different source schemas. This is done by exploiting knowledge in a Common Thesaurus together with a combination of clustering techniques and Description Logics. The approach is very similar to CUPID and presents the same drawbacks in matching hierarchical classifications. Furthermore, MOMIS includes an interactive process as a step of the integration procedure, and thus, unlike CTXMATCH, it does not support a fully automatic and run-time generation of mappings.

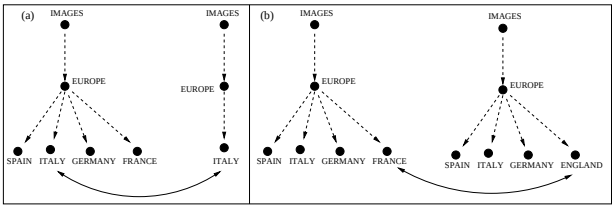

Figure 3. Example of right and wrong mapping

GLUE [18] is a taxonomy matcher that builds mappings taking advantage of information contained in instances, using machine learning techniques and domain-dependent constraints, manually provided by domain experts. GLUE represents an approach complementary to CTXMATCH. GLUE is more effective when a large amount of data is available, while CTXMATCH is more performant when less data are available, or the application requires a quick, on-the-fly mapping between structures. So, for instance, in case of product classification such as UNSPSC or Eclss (which are pure hierarchies of concepts with no data attached), GLUE cannot be applied. Combining the two approaches is a challenging research topic, which can probably lead to a more precise and effective methodology for semantic coordination.

6 CONCLUSIONS

In this paper we presented a new approach to semantic coordination in open and distributed environments, and an algorithm (called CTXMATCH) that implements this method for hierarchical classifications[16].

Furthermore, this approach to Distributed Knowledge Management has been applied to a series of practical applications. In particular, the EDAMOK (`http://edamok.itc.it`) project developed a P2P technology called KEEx (Knowledge Enhancement and Exchange), which is coherent with the vision of DKM. Indeed, P2P systems seem particularly suitable to implement a DKM system. In KEEx, each community is represented by a knowledge peer (K-peer), and a DKM system is implemented in a quite straightforward way: (i) each K-peer provides all the services needed by a knowledge node to create and organize its own local knowledge, and (ii) social structures and protocols of meaning negotiation are introduced to achieve semantic coordination (e.g., when searching documents from other peers – for more details see [6]). Throughout the EDAMOK project, KEEx has been applied and tested in several business cases such as healthcare [25], in which different actors, like hospital doctors, home

[15] The results of this experiment is not reported as they are not comparable with our simple keyword-based baseline, which makes no sense with multiple languages.

[16] CTXMATCH has been successfully tested on real HCs (i.e., pre-existing classifi cations used in real applications) and the results are described in [23].

	graph matching	CUPID	MOMIS	GLUE	CTXMATCH
Structural knowledge	•	•	•		•
Lexical knowledge		•	•	•	•
Knowledge base				•	•
Instance-based knowledge				•	
Type of result	Pairs of nodes	Similarity measure $\in [0..1]$ between pairs of nodes	Similarity measure $\in [0..1]$ between pairs of nodes	Similarity measure $\in [0..1]$ between pairs of nodes	Semantic relations between pairs of nodes

Table 2. Comparing CTXMATCH with other methods

doctors, nurses, technical and administrative staff and patients themselves, have to cooperate in order to achieve the main goal of healthcare organizations, that is, having patients well treated. Another interesting business case regards the B2B sector and, in particular, digital marketplaces and e-procurement [9]. In order to coordinate different procurement and supply processes, marketplace participants need to share heterogeneous information such as products and services catalogues. A third business case deals with inter-organizational cooperation and, in particular, refers to the Balearic Island Tourism system. Here the problem is how to support the collaboration among a set of public and private entities, all involved in promoting a sustainable development of tourism. Planning sustainable development means taking into account different information that belong to different domains such as economy, socio-politics, and ecology. For an extended analysis of these business cases see [10]. Currently KEEx has become a business application owned by Distributed Thinking (WWW.Dthink.biz), that is both using DKM technologies as an integral part of KM projects, and as an embedded P2P software in SUNs Star Office called PKM (Personal Knowledge Manager).

Furthermore, this approach is going to be applied in a peer-to-peer wireless system for ambient intelligence [15].

An important lesson we learned from this work is that methods for semantic coordinations should not be grouped together on the basis of the type of abstract structure they aim at coordinating (e.g., graphs, concept hierarchies), but on the basis of the intended use of the structures under consideration. In this paper, we addressed the problem of coordinating concept hierarchies when used to build hierarchical classifications. Other possible uses of structures are: conceptualizing some domain (ontologies), describing services (automata), describing data types (schemas). This "pragmatic" level (i.e., the use) is essential to provide the correct interpretation of a structure, and thus to discover the correct mappings with other structures.

The importance we assign to the fact that HCs are labelled with meaningful expressions does not mean that we see the problem of semantic coordination as a problem of natural language processing (NLP). On the contrary, the solution we provided is mostly based on knowledge representation and automated reasoning techniques. However, the problem of semantic coordination is a fertile field for collaboration between researchers in knowledge representation and in NLP. Indeed, if in describing the general approach one can assume that some linguistic meaning analysis for labels is available and ready to use, we must be very clear about the fact that real applications (like the one we described in Section 3) require a massive use of techniques and tools from NLP, as a good automatic analysis of labels from a linguistic point of view is a necessary precondition for applying the algorithm to HC in local applications, and for the quality of mappings resulting from the application of the algorithm.

The work we presented in this paper is only the first step of a very ambitious scientific challenge, namely to investigate what is the minimal common ground needed to enable communication between autonomous entities (e.g., agents) that cannot look into each others head, and thus can achieve some degree of semantic coordination only through other means, like exchanging examples, pointing to things, remembering past interactions, generalizing from past communications, and so on. To this end, a lot of work remains to be done. On our side, the next steps will be: extending the algorithm beyond classifications (namely to structures with purposes other than classifying things, as for example catalogues); generalizing the types of structures we can match (for example, structures with non hierarchical relations, e.g. roles); going beyond WORDNET as a source of lexical and domain knowledge; allowing different lexical and/or domain knowledge sources for each of the local structures to be coordinated, migrating from propositional logics to description logics for a more powerful expressivity. The last problem is perhaps the most challenging one, as it introduces a situation is which the space of 'senses' is not necessarily shared, and thus we cannot rely on that information for inferring a semantic relation between labels of distinct structures.

REFERENCES

[1] A. Abecker, A. Bernardi, L. Elst, R. Herterich, C. Houy, S. Mller, S. Dioudis, G. Mentzas, and M. Legal, 'Workfbw-embedded organizational memory access: The decor project', *IJCAI'2001 Working Notes of the Workshop on Knowledge Management and Organizational Memories, Seattle, Washington, USA, pp. 1-9*, (August 2001).

[2] A. Abecker, A. Bernardi, K. Hinkelman, O. Khn, and Michael Sintek, 'Context-aware, proactive delivery of task-specifi c knowledge: The knowmore project', *Int. Journal on Information Systems Frontiers (ISF)2(3/4):139-162,Special Issue on Knowledge Management and Organizational Memory, Kluwer*, (2000).

[3] C. Argyris and D.A. Schon, *Organisational Learning II: Theory, Method, and Practice*, Addison-Wesley, 2002.

[4] Sonia Bergamaschi, Silvana Castano, and Maurizio Vincini, 'Semantic integration of semistructured and structured data sources', *SIGMOD Record*, **28**(1), 54–59, (1999).

[5] J.R. Boland and R.V.Tenkasi, 'Perspective making and perspective taking in communities of knowing', *Organization Science*, **6**(4), 350–372, (1995).

[6] M. Bonifacio, P. Bouquet, G. Mameli, and M. Nori, 'Kex: a peer-to-peer solution for distributed knowledge management', in *Fourth International Conference on Practical Aspects of Knowledge Management (PAKM-2002)*, eds., D. Karagiannis and U. Reimer, Vienna (Austria), (2002).

[7] M. Bonifacio, P. Bouquet, and P. Traverso, 'Enabling distributed knowledge management. managerial and technological implications', *Novatica and Informatik/Informatique*, **III**(1), (2002).

[8] M. Bonifacio, P. Camussone, and C. Zini, 'Managing the km trade-off: Knowledge: Centralization versus distribution', in *To be published in Journal of Universal Computer Science*.

[9] M. Bonifacio, A. Donà, A. Molani, and L. Serafi ni, 'Context matching for electronic marketplaces a case study', Technical report, Technical Report ITC-Irst, (March 2003).

[10] M. Bonifacio and A. Molani, 'The richness of diversity in knowledge creation: an interdisciplinary overview', in *Journal of Universal Computer Science, 9(6)*, (2003).

[11] P. Bouquet, L. Serafi ni, and S. Zanobini, 'Semantic coordination: a new approach and an application', in *Second International Semantic Web Conference (ISWC-03)*, ed., K. Sycara, Lecture Notes in Computer Science (LNCS), Sanibel Island (Florida, USA), (October 2003).

[12] J. S. Brown and P. Duguid, 'Knowledge and organization: A social–practice perspective', *Science*, (2001).

[13] S.J. Brown and P. Duguid, 'Organizational learning and communities-of-practice : Toward a unifi ed view of working, learning and innovation', *Organization Science*, **2**(1), (1991).

[14] A. Büchner, M. Ranta, J. Hughes, and M. Mäntylä, 'Semantic information mediation among multiple product ontologies', in *Proc. 4th World Conference on Integrated Design & Process Technology*, (1999).

[15] P. Busetta, P. Bouquet, G. Adami, M. Bonifacio, and F. Palmieri, 'K-Trek: An approach to context awareness in large environments', Technical report, Istituto per la Ricerca Scientifi ca e Tecnologica (ITC-IRST), Trento (Italy), (April 2003). Submitted to UbiComp'2003.

[16] Jeremy Carroll and Hewlett-Packard, 'Matching rdf graphs', in *Proc. in the first International Semantic Web Conference - ISWC 2002*, pp. 5–15, (2002).

[17] T.H. Davenport and L. Prusak, *Working Knowledge*, Harvard Business School Press, 2000.

[18] A. Doan, J. Madhavan, P. Domingos, and A. Halevy, 'Learning to map between ontologies on the semantic web', in *Proceedings of WWW-2002, 11th International WWW Conference, Hawaii*, (2002).

[19] P. Shvaiko F. Giunchiglia, 'Semantic matching', *Proceedings of the workshop on Semantic Integration*, (October 2003).

[20] *WordNet: An Electronic Lexical Database*, ed., Christiane Fellbaum, The MIT Press, Cambridge, US, 1998.

[21] J. Lave and E. Wenger, *Situated Learning: Legitimate Peripheral Participation*, Cambridge University Press, New York, 1990.

[22] Jayant Madhavan, Philip A. Bernstein, and Erhard Rahm, 'Generic schema matching with cupid', in *The VLDB Journal*, pp. 49–58, (2001).

[23] B. M. Magnini, L. Serafi ni, A. Doná, L. Gatti, C. Girardi, , and M. Speranza, 'Large–scale evaluation of context matching', Technical Report 0301–07, ITC–IRST, Trento, Italy, (2003).

[24] Tova Milo and Sagit Zohar, 'Using schema matching to simplify heterogeneous data translation', in *Proc. 24th Int. Conf. Very Large Data Bases, VLDB*, pp. 122–133, (24–27 1998).

[25] A. Molani, A. Perini, E. Yu, and P. Bresciani, 'Analysing the requirements for knowledge management using intentional requirements', in *to appear in Proceedings AAAI Spring Symposium on Agent Mediated Knowledge Management (AMKM-03), Stanford, USA, March 24-26*, (2003).

[26] Marcello Pelillo, Kaleem Siddiqi, and Steven W. Zucker, 'Matching hierarchical structures using association graphs', *Lecture Notes in Computer Science*, **1407**, (1998).

[27] Jason Tsong-Li Wang, Kaizhong Zhang, Karpjoo Jeong, and Dennis Shasha, 'A system for approximate tree matching', *Knowledge and Data Engineering*, **6**(4), 559–571, (1994).

[28] E.K. Weick, 'What theory is not, theorizing is', *Adminstrative Science Quarterly*, **40**, (1995).

[29] E. Wenger, *Communities of Practice. Learning, Meaning, and Identity*, Cambridge University Press, 1998.

[30] K. Zhang, J. T. L. Wang, and D. Shasha, 'On the editing distance between undirected acyclic graphs and related problems', in *Proceedings of the 6th Annual Symposium on Combinatorial Pattern Matching*, eds., Z. Galil and E. Ukkonen, volume 937, pp. 395–407, Espoo, Finland, (1995). Springer-Verlag, Berlin.