Semantic coordination of heterogeneous classifications schemas

Paolo Bouquet[1], Luciano Serafini[2], and Stefano Zanobini[1]

[1] University of Trento, Italy {bouquet,zanobini}@dit.unitn.it
[2] IRST, Trento, Italy serafini@itc.it

1 Introduction

One of the key challenges in the development of open distributed systems is enabling the exchange of meaningful information across applications which (i) may use autonomously developed schemas for organizing locally available data, and (ii) need to discover relations between schemas 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[3]. 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), this approach can't work for several 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. 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 autonomously developed schemas and possibly heterogeneous ontologies) discover relations across their schemas and use them to provide the required services.

In this paper, we propose a general approach to the problem of discovering mappings across the schemas of two or more semantic peers. The method we propose is *intrinsically semantic*, as the mappings it discovers between nodes of different schemas are computed as a logical consequence of (1) the explicit representation of the meaning of each node in the schemas, and (2) additional background knowledge (if available). The method is illustrated and tested on a significant instance of the

[3] But see [3] for a discussion of the drawbacks of this approach from the standpoint of Knowledge Management applications.

problem, namely the problem of matching hierarchical classifications (HCs). The main technical contribution of this part is an algorithm, called CTXMATCH, which takes in input two HCs H and H' and, for each pair of concepts $k \in H$ and $k' \in H'$, returns their semantic relation (called a mapping).

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 elements belonging to different schemas are deduced via logical reasoning, rather then derived through (more or less) complex heuristic techniques, and thus can be assigned a clearly defined model-theoretic semantics. This shifts the problem of semantic coordination from the problem of computing linguistic or structural similarities between schemas to the problem of deducing relations between formulas that represent the meaning of each concept in a schema. This explains, for example, why our approach performs much better than most heuristic-based methods when two nodes intuitively represent equivalent concetps, but occur in classification schemas which are structurally very different.

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 CTXMATCH, the proposed algorithm for coordinating HCs. Finally, we compare our approach with other proposed approaches for matching schemas (Section 4).

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 this sense, each peer is a semantically autonomous agent). Each peer can organize data using one or more schemas (e.g., database schemas, directory trees in a file system, classification schemas, taxonomies, and so on). Different peers may use different schemas to organize the same collection of data, and conversely the same schema can be used to organize different collections of data.

We also assume that semantic peers need to exchange data (e.g. documents classified under different classification schemas) to perform 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

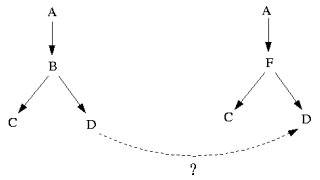

Fig. 1. Mapping abstract structures

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.

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 this 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

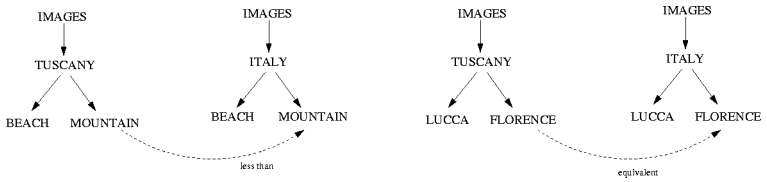

Fig. 2. Mapping schemas with meaningful labels

words and more complex 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 elicitation of the meaning associated to each node in a schema[4]. The semantic elicitation process may require the use if three different levels of knowledge:

Lexical knowledge: 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), or the fact that 'bachelor' and 'unmarried man' can mean the same thing (synonymy);

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

Structural knowledge: knowledge which derives 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'. World knowledge 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) elicit the meaning expressed by the left hand node, ('images of Tuscan mountains', denoted by P), (ii)

[4]Even though a discussion on the difference between schemas and ontologies is beyond the scope of this paper, notice that schemas – such as the two classifications in Figure 2 – cannot be viewed as straight ontologies – not even *lightweight* ontologies, as the information they convey is mostly implicit in their labels, and in a body of knowledge associated with labels. Indeed, we can say that a classification schema, as many other types of schemas, is a very concise way of referring to complex concepts (like "images of Tuscan montains"), but the identification of the concept corresponding to each element in a schema may require a lot of semantic and world knowledge, which can only be made available to computer programs via explicit semantic models (ontologies). See the rest of the paper for a practical illustration of this point

elicit the meaning expressed by the right hand node ('images of Italian mountains', denoted by P'), and finally (iii) derive the semantic relation between the meaning of the two nodes, namely that $P \sqsubseteq P'$ (intuitively, subsumption between the concepts corresponding to the two schema elements).

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 [5] used for grouping/organizing/classifying data (such as documents, goods, activities, services) in categories. Some well-known examples of HCs are web directories (see e.g. the Google$^{\mathrm{TM}}$ Directory or the Yahoo!$^{\mathrm{TM}}$Directory), file systems, document databases, ...

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 some representation O of world knowledge.*

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 world knowledge. 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 [9]. 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, like OWL, Prolog clauses, DAML/OIL, RDFS.

Our method is designed for scenarios in which an agent A (called the *seeker*) needs to find new documents relative to some category in its local HC S_A. Imagine that another agent B (called the *provider*) owns a collection of potentially relevant documents, but they are classified using a different HC S_B. Our problem is to discover semantic relations between A's original category in S_A and the categories in S_B, and – based on the discovered relations – return the relevant documents. A and B are called semantic peers because, as we will say, each of them has equivalent capabilities and responsibilities in assigning meaning to the classification schemas used to organize local documents, and in assigning documents to categories.

A collection of point-to-point relations between categories of two distinct HCs is called a mapping:

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

In this version of the algorithm, five relations are allowed between the concepts corresponding to two nodes belonging to different HCs: $m \sqsupseteq n$ (m is *more general than* n); $m \sqsubseteq n$ (m is *less general than* n); $m \equiv n$ (m is *equivalent* to n); $m \sqcap n$ is consistent (i.e. it has an interpretation which is not empty and thus m is *compatible* with n); $m \sqcap n \sqsubseteq \bot$ (m is *disjoint* from n).

The algorithm CTXMATCH takes as **inputs** the seeker's and the provider's classification schemas S and S', and the provider's lexicon L and knowledge base O[5]. As it will become clear in what follows, this means that the resulting mapping is *directional*, as it represents the provider's point of view on the relation between S and S' (because it is based on the provider's lexicon and knowledge base, and thus on the provider's understanding of the two schemas). As the seeker in principle may use different lexical and background knowledge, a different mapping between the same schemas S and S' might be computed. The reason why here we privilege the provider's perspective is that it reflects the scenario in which an agent asks for information to one or more agents, whose answers are necessarily based on their understanding of the question.

The **output** of the algorithm is a mapping $\mathcal{M}$.

[5] In the version of the algorithm presented here, we use WORDNET both as a source of lexical and world knowledge. However, WORDNET can be replaced by any other combination of a lexical and a world knowledge source.

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 $m \in S$ and $n \in S'$ **do**
2 $\langle \phi, \Theta \rangle \leftarrow$ SEMANTIC−ELICITATION(m, S, L, O);
3 $\langle \psi, \Upsilon \rangle \leftarrow$ SEMANTIC−ELICITATION(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 elicitation*, 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 relevant knowledge about it (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). In our framework, the pair $\langle \phi, \Theta \rangle$, called *contextualized concept*, expresses 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−ELICITATION and SEMANTIC−COMPARISON.

3.1 Semantic elicitation

In this phase we make explicit in a logical formula[6] the meaning of a node n in a HC S.

[6] The choice of the formal language 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 this implementation we adopt the propositional fragment of Description logics, where each propositional letter corresponds to a concept (synset) provided by WORDNET. However, in [18], a richer encoding is described which uses also the DL roles. As an example, the node `MOUNTAIN` of the left hand schema of Figure 2, now interpreted as

Algorithm 2 SEMANTIC–ELICITATION(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 Step 1 and Step 2, the function EXTRACT–CANDIDATE–CONCEPTS uses lexical knowledge to associate to each word occurring in the nodes of an HC all concepts possibly 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[7].

In 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 world knowledge, we can discover, for example, that '`florence#1` PartOf `tuscany#1`', i.e. that there exists a 'part of' relation between the first sense of 'Florence' and the first sense of 'Tuscany'. World knowledge relations extracted from WORDNET are translated into logical axioms according to Table 1. So, the relation '`florence#1` PartOf `tuscany#1`' is encoded as '$\texttt{florence\#1} \sqsubseteq \texttt{tuscany\#1}$'[8].

(`image#1` $\sqcup \ldots \sqcup$ `image#8`) $\sqcap$ `tuscany#1` $\sqcap$ `mountain#1`, is encoded as (`image#1` $\sqcup \ldots \sqcup$ `image#8`) $\sqcap \exists$`about#3`.(`mountain#1` $\sqcap \exists$`locatedIn#2`.`tuscany#1`), namely 'images about mountains that are located in Tuscany'.

[7] 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.

[8] For heuristical reasons – see [4] – we consider only relations between concepts on the same path of a HC and their siblings.

WORDNET World knowledge relations	axiom
s#k synonym t#h	s#k $\equiv$ t#h
s#k { hyponym | PartOf } t#h	s#k $\sqsubseteq$ t#h
s#k { hypernym | HasPart } t#h	t#h $\sqsubseteq$ s#k
s#k antonym t#h	(t#k $\sqcap$ s#h) $\sqsubseteq \bot$

Table 1. WORDNET relations and corresponding axioms.

Step 4 has the goal of filtering out unlikely senses associated to a node's label. Going back to the previous example, we tentatively discard one of the senses associated to the node FLORENCE ('a town in northeast South Carolina', florence#2), based on the fact that we found the local axiom 'florence#1 PartOf tuscany#1' which links the other sens of 'Florence' to a sense of 'Tuscany'. This fact is used to make the conjecture that the contextually relevant sense of Florence is the city in Tuscany, and not the city in the USA. When ambiguity persists (because there are axioms related to different senses, or no axioms at all), all possible senses are kept 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 [4], node labels are first processed one by one to build a preliminary interpretation, called *simple concept*, which doesn't take into account the position of the node in the structure. For example, the simple concept associated to the node FLORENCE of the left hand structure of Figure 2 is the atom florence#1 (i.e. one of the two senses provided by WORDNET and not discarded by filtering). Then, these results are combined for generating a formula approximating the meaning expressed by a node *in a schema*. 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 the root node to n. So, the formula approximating the meaning expressed by the node FLORENCE in that HC is (image#1 $\sqcup \ldots \sqcup$ image#8) $\sqcap$ tuscany#1 $\sqcap$ florence#1.

Step 6 returns the formula expressing the meaning of the node and the set of local axioms found in Step 3. This formula represents what we call a contextualized concept, namely a complex concept associated to a node in a schema, given L and O.

This explains why the set of contextualized concepts extracted from a HC can be viewed as a *context* in the sense of [11, 1], namely a partial and approximate representation of the world from an individual's perspective. Indeed, it reflects a semantic peer's perspective on a collection of documents. As we already pointed out, the same schema can be transformed into a different context by different semantic peers, as they might use a different lexicon L' or different world knowledge O'. This explains why the mappings between the same pair of schemas computed by different peers are not identical.

3.2 Semantic comparison

The goal of this phase is to find the semantic relation which holds between two contextualized concepts (associated to two nodes in different HCs).

In Step 1, the function EXTRACT–RELATIONAL–AXIOMS tries to find axioms which connect concepts belonging to different HCs. This function is similar to EXTRACT–LOCAL–AXIOMS in the semantic elicitation part. 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' (`tuscany#1` $\sqsubseteq$ `italy#1`).

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

VarDeclarations
set of formulas Γ
semantic relation R

1 $\Gamma \leftarrow$ EXTRACT–RELATIONAL–AXIOMS(ϕ, ψ, O);
2 **if** $\Theta,\Upsilon,\Gamma \models (\phi \sqcap \psi) \sqsubseteq \bot$ **then** $R \leftarrow disjoint$;
3 **else if** $\Theta,\Upsilon,\Gamma \models (\phi \equiv \psi)$ **then** $R \leftarrow equivalent$;
4 **else if** $\Theta,\Upsilon,\Gamma \models (\phi \sqsubseteq \psi)$ **then** $R \leftarrow less\ general\ than$;
5 **else if** $\Theta,\Upsilon,\Gamma \models (\psi \sqsubseteq \phi)$ **then** $R \leftarrow more\ general\ than$;
6 **else** $R \leftarrow compatible$;
7 **Return** R;

In steps 2–6, 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. 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	`florence#1` $\sqsubseteq$ `tuscany#1`
ϕ	(`image#1` $\sqcup \ldots \sqcup$ `image#8`) $\sqcap$ `tuscany#1` $\sqcap$ `florence#1`
Δ	`florence#1` $\sqsubseteq$ `italy#1`
ψ	(`image#1` $\sqcup \ldots \sqcup$ `image#8`) $\sqcap$ `italy#1` $\sqcap$ `florence#1`
Γ	`tuscany#1` $\sqsubseteq$ `italy#1`

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

Θ	$\emptyset$
ϕ	(`image#1` $\sqcup \ldots \sqcup$ `image#8`) $\sqcap$ `tuscany#1` $\sqcap$ `mountain#1`
Δ	$\emptyset$
ψ	(`image#1` $\sqcup \ldots \sqcup$ `image#8`) $\sqcap$ `italy#1` $\sqcap$ `mountain#1`
Γ	`tuscany#1` $\sqsubseteq$ `italy#1`

The returned relation is '$less\ general\ than$'.

Following on the idea that a semantically elicited schema is a context, a mapping between two contextualized concepts belonging to different contexts can be formally represented as a *compatibility relation* [10], namely a constraint on the local models of the two contexts. In this sense, the algorithm we present is a first attempt to discover (rather than assume) relations over local models of two or more contexts (which, from a proof–theoretical point of view, corresponds to discover bridge rules [12] across contexts).

4 Related Work

Recently, many methods have been proposed for matching heterogeneous schemas (see [17] for a well-known survey of the area). However, our method shifts the problem of semantic coordination from the problem of matching (in a more or less sophisticated way) schemas to the problem of inferring semantic relations between the meaning of schema elements. Under this respect, to the best of our knowledge, our approach is still original; an alternative (and partially extended) implementation, called S-Match, was proposed in [8] as a rationalization of CTXMATCH. Therefore, a straightforward comparison with other methods is not easy.

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.

	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

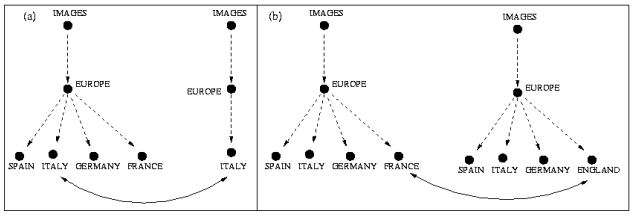

Fig. 3. Example of right and wrong mapping

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 [20, 19, 16, 15, 6].

CUPID [13] 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) [2] 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.

GLUE [7] 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.

5 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. CTXMATCH has been successfully tested on real HCs (i.e., pre-existing classifications used in real applications) and the results are described in [14].

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 deal with (e.g., DAGs, concept hierarchies), but on the basis of the intended use of the structures under consideration. In this paper, we addressed the problem of semantic coordination for hierarchical classifications, and the elicitation method we proposed heavily relies on this assumption. However, there are other possible uses for "similar" structures, e.g. specifying the conceptualization of some domain (ontologies), describing web services (finite automata), describing data types (schemas). This "pragmatic" level (i.e., the use of a schema) 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 analysis on labels is available and ready to use, real applications require a massive use of techniques and tools from NLP, as a reliable, automatic analysis of labels from a linguistic point of view is a necessary precondition for the quality of the algorithm's results.

The work we presented 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 that cannot look into each others head, and thus can achieve some degree of semantic coordination only through

other means, like exchanging messages, passing 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. In particular, we stress that CTXMATCH is a one-shot method for discovering relations across static schemas; however, much more interesting is the problem of dynamically adapting the schemas (or their interpretation) as a result of the interaction between two (or more) semantic peers. This much more general task is what we call meaning negotiation (as opposed to meaning coordination), where peers may try to find an agreement on the meaning of schemas, or even update/change their lexical/background knowledge to achieve a more satisfactory mapping with other peers. This project seems quite exciting and challenging, as it requires to go beyond pure meaning, and take into account other dimensions like a cost/benefit analysis of changing a schema, updating/changing a body of lexical and/or background knowledge, redefining mappings across schemas. This *economics of meaning* is a completely new field, whose crucial relevance for large-scale projects (like the Semantic Web) and semantic-based applications have been recognized only recently, and where new models and tools need to be developed from scratch.

References

1. M. Benerecetti, P. Bouquet, and C. Ghidini. Contextual Reasoning Distilled. *Journal of Theoretical and Experimental Artificial Intelligence*, 12(3):279–305, 2000.
2. S. Bergamaschi, S. Castano, and M. Vincini. Semantic integration of semistructured and structured data sources. *SIGMOD Record*, 28(1):54–59, 1999.
3. M. Bonifacio, P. Bouquet, and P. Traverso. Enabling distributed knowledge management. managerial and technological implications. *Novatica and Informatik/Informatique*, III(1), 2002.
4. P. Bouquet, L. Serafini, and S. Zanobini. Semantic coordination: a new approach and an application. In K. Sycara, editor, *Second International Semantic Web Conference (ISWC-03)*, Lecture Notes in Computer Science (LNCS), Sanibel Island (Florida, USA), October 2003.
5. 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.
6. Jeremy Carroll and Hewlett-Packard. Matching rdf graphs. In *Proc. in the first International Semantic Web Conference - ISWC 2002*, pages 5–15, 2002.
7. 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.
8. F. Giunchiglia, P. Shvaiko, M. Yatskevich. S-Match: an Algorithm and an Implementation of Semantic Matching. In *Proceedings of the first European Semantic Web Symposium* (ESWC-2004), Springer, LNCS 3053, pp. 61-75.
9. Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. The MIT Press, Cambridge, US, 1998.
10. C. Ghidini and F. Giunchiglia. Local Models Semantics, or Contextual Reasoning = Locality + Compatibility. *Artificial Intelligence*, 127(2):221–259, 2001.

11. F. Giunchiglia. Contextual reasoning. *Epistemologia, special issue on I Linguaggi e le Macchine*, XVI:345–364, 1993. Short version in Proceedings IJCAI'93 Workshop on Using Knowledge in its Context, Chambery, France, 1993, pp. 39–49. Also IRST-Technical Report 9211-20, IRST, Trento, Italy.
12. F. Giunchiglia and L. Serafini. Multilanguage hierarchical logics or: how we can do without modal logics. *Artificial Intelligence*, 65(1):29–70, 1994. Also IRST-Technical Report 9110-07, IRST, Trento, Italy.
13. Jayant Madhavan, Philip A. Bernstein, and Erhard Rahm. Generic schema matching with cupid. In *The VLDB Journal*, pages 49–58, 2001.
14. B. M. Magnini, L. Serafini, A. Doná, L. Gatti, C. Girardi, , and M. Speranza. Large–scale evaluation of context matching. Technical Report 0301–07, ITC–IRST, Trento, Italy, 2003.
15. Tova Milo and Sagit Zohar. Using schema matching to simplify heterogeneous data translation. In *Proc. 24th Int. Conf. Very Large Data Bases, VLDB*, pages 122–133, 24–27 1998.
16. Marcello Pelillo, Kaleem Siddiqi, and Steven W. Zucker. Matching hierarchical structures using association graphs. Springer, LNCS 1407, 1998.
17. E. Rahm, P.A. Bernstein. A survey of approaches to automatic schema matching. *VLDB Journal*, 10(4), 2001.
18. S. Sceffer, L. Serafini, S. Zanobini. Semantic coordination of hierarchical classifications with attributes. Technical Report 706, University of Trento, Italy, December 2004. `http://eprints.biblio.unitn.it/archive/00000706/`
19. 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.
20. K. Zhang, J. T. L. Wang, and D. Shasha. On the editing distance between undirected acyclic graphs and related problems. In Z. Galil and E. Ukkonen, editors, *Proceedings of the 6th Annual Symposium on Combinatorial Pattern Matching*, volume 937, pages 395–407, Espoo, Finland, 1995. Springer.