Semantic coordination: a new approach and an application

P. Bouquet[1,2], L. Serafini[2], and S. Zanobini[1]

[1]Department of Information and Communication Technology – University of Trento
Via Sommarive, 10 – 38050 Trento (Italy)
[2]ITC-IRST – Istituto per la Ricerca Scientifica e Tecnologica
Via Sommarive, 14 – 38050 Trento (Italy)
bouquet@dit.unitn.it serafini@itc.it zanobini@dit.unitn.it

Abstract. Semantic coordination, namely the problem of finding an agreement on the meaning of heterogeneous semantic models, is one of the key issues in the development of the Semantic Web. In this paper, we propose a new algorithm for discovering semantic mappings across hierarchical classifications based on a new approach to semantic coordination. 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 formulae that represent the meaning of concepts belonging to different models. We show how to apply the approach and the algorithm to an interesting family of semantic models, namely hierarchical classifications, and present the results of preliminary tests on two types of hierarchical classifications, web directories and catalogs. Finally, we argue why this is a significant improvement on previous approaches.

1 Introduction

One of the key issues in the development of the Semantic Web is the problem of enabling machines to exchange meaningful information/knowledge across applications which (i) may use autonomously developed models of locally available data (local models), and (ii) need to find a sort of agreement on what local models are about to achieve their users' goals. This problem can be viewed as a problem of *semantic coordination*[1], defined as follows: (i) all parties have an interest in finding an agreement on how to map their models onto each others, but (ii) there are many possible/plausible solutions (many alternative mappings across local models) among which they need to select the right, or at least a sufficiently good, one.

In environments with more or less well-defined boundaries, like a corporate Intranet, the semantic coordination problem can be addressed by defining and using shared models (e.g., ontologies) throughout the entire organization[2]. However, in open environments, like the Semantic Web, this "centralized" approach to semantic coordination is

[1] See the introduction of [6] for this notion, and its relation with the notion of *meaning negotiation*.

[2] But see [4] for a discussion of the drawbacks of this approach from the standpoint of Knowledge Management applications.

not viable for several reasons, such as the difficulty of "negotiating" a shared model of data that suits the needs of all parties involved, the practical impossibility of maintaining such a model in a highly dynamic environment, the problem of finding a satisfactory mapping of pre-existing local models onto such a global model. In such a scenario, the problem of exchanging meaningful information across locally defined models seems particularly tough, as we cannot presuppose an *a priori* agreement, and therefore its solution requires a more dynamic and flexible form of "peer-to-peer" semantic coordination.

In this paper, we address an important instance of the problem of semantic coordination, namely the problem of coordinating hierarchical classifications (HCs). HCs are structures having the *explicit* purpose of organizing/classifying some kind of data (such as documents, records in a database, goods, activities, services). The problem of coordinating HCs is significant for at least two main reasons:

- first, HCs are widely used in many applications[3]. Examples are: web directories (see e.g. the Google™ Directory or the Yahoo!™Directory), content management tools and portals (which often use hierarchical classifications to organize documents and web pages), service registry (web services are typically classified in a hierarchical form, e.g. in UDDI), marketplaces (goods are classified in hierarchical catalogs), PC's file systems (where files are typically classified in hierarchical folder structures);
- second, it is an empirical fact that most actual HCs (as most concrete instances of models available on the Semantic Web) are built using structures whose labels are expressions from the language spoken by the community of their users (including technical words, neologisms, proper names, abbreviations, acronyms, whose meaning is shared in that community). In our opinion, recognizing this fact is crucial to go beyond the use of syntactic (or weakly semantic) techniques, as it 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 of a HC are taken.

The main technical contribution of the paper is a logic–based algorithm, called CTXMATCH, for coordinating HCs. It 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. The relations we consider in this version of CTXMATCH are: k is less general than k', k is more general than k', k is equivalent to k', k is compatible with k', and k is incompatible with (i.e., disjoint from) k'. The formal semantics of these relations will be made precise in the paper.

With respect to other approaches to semantic coordination proposed in the literature (often under different "headings", such as schema matching, ontology mapping, semantic integration; see Section 6 for references and a detailed discussion of some of them), our approach is innovative in three main aspects: (1) we introduce a new method for making explicit the meaning of nodes in a HC (and in general, in structured semantic models) by combining three different types of knowledge, each of which has a specific role; (2) the result of applying this method is that we are able to produce a

[3] For an interesting discussion of the central role of classification in human cognition see, e.g., [15, 7].

new representation of a HC, in which all relevant knowledge about the nodes (including their meaning in that specific HC) is encoded as a set of logical formulae; (3) mappings across nodes of two HCs are then 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. As we will show, this leads to a major conceptual shift, as the problem of semantic coordination between HCs is no longer tackled as a 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), but rather as a problem of deducing relations between formulae that represent the meaning of each concept in a given HC. 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 show how this approach is instantiated to the problem of coordinating HCs. Then we present the main features of CTXMATCH the proposed algorithm for coordinating HCs (Section 4). In the final part of the paper, we sum-up the results of testing the algorithm on web directories and catalogs (Section 5) and compare our approach with other proposed approaches for matching schemas (Section 6).

2 Our approach

The approach to semantic coordination we propose in this paper is based on the intuition that there is an essential conceptual difference between coordinating generic abstract structures (e.g., arbitrary labelled graphs) and coordinating structures whose labels are taken from the language spoken by the community of their users. Indeed, the second type of structures give 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. Most importantly, 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 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 HCs.

To clarify this intuition, let us consider the HCs in Figure 1, and suppose they are used to classify images in two multi-media repositories. Imagine we want to discover the semantic relation between the nodes labelled `MOUNTAIN` in the two HCs on the left hand side, and between the two nodes `FLORENCE` on the right hand side. Using knowledge about the meaning of labels and about the world, we understand almost immediately that the relation between the first pair of nodes is "less general than" (intuitively, the images that one would classify as images of mountains in Tuscany is a subset of images that one would classify under images of mountains in Italy), and that the relation between the second pair of nodes is "equivalent to" (the images that one would classify as images of Florence in Tuscany are the same as the images that one would classify under images of Florence in Italy). Notice that the relation is different, even

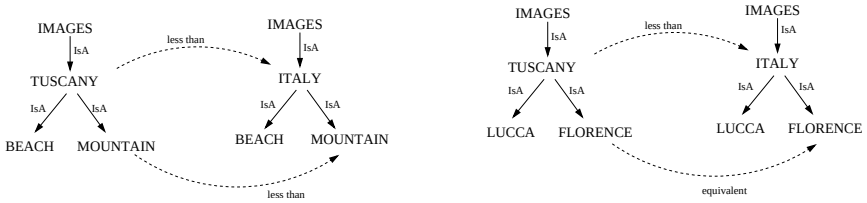

Fig. 1. Coordinating HCs

though the two pairs of HCs are structurally very similar. How do we design a technique of semantic coordination which exploits the same kind of facts to achieve the same results?

The approach we propose is based on three basic ideas. First of all, exploiting the degree of coordination implicit in the fact that labels are taken from language requires to make explicit the meaning of labels associated to each node in a HC. We claim that this can be done only of we properly take into account three distinct levels of semantic knowledge:

Lexical knowledge: knowledge about the words used in the labels. For example, the fact that the word 'image' can be used in the sense of a picture or in the sense of personal facade, and the fact that different words may have the same sense (e.g., 'picture' and 'image');

Domain knowledge: knowledge about the relation between the senses of labels in the real world or in a specific domain. For example, the fact that Tuscany is part of Italy, or that Florence is in Italy;

Structural knowledge: knowledge deriving from how labels are arranged in a given HC. For example, the fact that the concept labelled `MOUNTAIN` classifies images, and not books.

Let us see how these three levels can be used to explain the intuitive reasoning described above. Consider the mapping between the two nodes `MOUNTAIN`. Linguistic meaning can be used to assume that the sense of the two labels is the same. Domain knowledge tells us, among other things, that Tuscany is part of Italy. Finally, structural knowledge tells us that the intended meaning of the two nodes `MOUNTAIN` is images of Tuscan mountains (left HC) and images of Italian mountains (right HC). All these facts together allow us to conclude that one node is less general than the other one. We can use similar reasoning for the two nodes `FLORENCE`, which are structurally equivalent. But exploiting domain knowledge, we can add the fact that Florence is in Tuscany (such a relation doesn't hold between mountains and Italy in the first example). This further piece of domain knowledge allows us to conclude that, beyond structural similarity, the relation is different.

Second, this analysis of meaning has an important consequence on our approach to semantic coordination. Indeed, unlike all other approaches we know of, we do not use lexical knowledge (and, in our case, domain knowledge) to improve the results

of structural matching (e.g., by adding synonyms for labels, or expanding acronyms). Instead, we combine knowledge from all three levels to build a new representation of the problem, where the meaning of each node is encoded as a logical formula, and relevant domain knowledge and structural relations between nodes are added to nodes as sets of axioms that capture background knowledge about them.

This, in turn, introduces the third innovative idea of our approach. Indeed, once the meaning of each node, together with all relevant domain and structural knowledge, is encoded as a set of logical formulae, the problem of discovering the semantic relation between two nodes can 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 Section 4, determining whether there is an equivalence relation between the meaning of two nodes becomes 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 Semantic coordination of hierarchical classification

In this section we show how to apply the general approach described in the previous section to the problem of coordinating HCs. Intuitively, a classification is a grouping of things into classes or categories. When categories are arranged into a hierarchical structure, we have a hierarchical classification. Formally, the hierarchical structures we use to build HCs are *concept hierarchies*, defined as follows in [8]:

Definition 1 (Concept hierarchy). *A* concept hierarchy *is a triple $H = \langle K, E, l \rangle$ where K is a finite set of nodes, E is a set of arcs on K, such that $\langle K, E \rangle$ is a rooted tree, and l is a function from $K \cup E$ to a set L of labels.*

Given a concept hierarchy H, a classification can be defined as follows:

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

We assume that the classification function μ in Definition 2 satisfies the following *specificity principle*: an object $d \in D$ is classified under a category k, if d is about k (according to the some criteria, e.g., the semantic intuition of the creator of the classification!) and there isn't a more specific concept k' under which d could be classified[4].

Prototypical examples of HCs are the web directories of many search engines, for example the Google™ Directory, the Yahoo!™ Directory, or the Looksmart™ web directory. A tiny fraction of the HCs corresponding to the Google™ Directory™ and to the Yahoo!™Directory is depicted in Figure 2.

[4] See for example Yahoo!™ instruction for "Finding an appropriate Category" at http://docs.yahoo.com/info/suggest/appropriate.html.

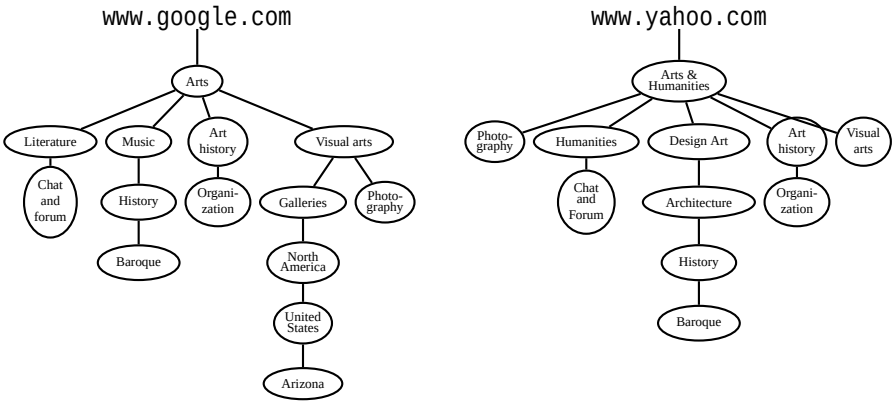

Fig. 2. Examples of concept hierarchies (source: Open Directory and Yahoo!Directory)

Intuitively, the problem of semantic coordination arises when one needs to find relations between categories belonging to distinct (and thus typically heterogeneous) HCs. Imagine the following scenario. You are browsing the Google^{TM} Directory on the left hand side of Figure 2, and find out that the documents classified under the category labelled `Baroque` are very relevant for your work on Baroque music. So you would like to ask the system to find out for you whether there are categories in different hierarchical classifications (e.g., the Yahoo!^{TM}Directory) which have the same meaning as, or a meaning related to, the category `Baroque` in the directory you are currently browsing[5]. Formally, we define the problem of semantic coordination as the problem of discovering mappings between categories in two distinct concept hierarchies H and H':

Definition 3 (Mapping). *A* mapping *M from $H = \langle K, E, l \rangle$ to $H' = \langle K', E', l' \rangle$ is a function $M : K \times K' \rightarrow rel$, where rel is a set of symbols, called the* possible relations.

The set *rel* of possible relations depends on the intended use of the structures we want to map. Indeed, in our experience, the intended use of a structure (e.g., classifying objects) is semantically much more relevant than the type of abstract structures involved to determine how a structure should be interpreted. As the purpose of mapping HCs is to discover relations between nodes (concepts) that are used to classify objects, five possible relations can hold between two nodes k_s and k_t belonging to different HCs: $k_s \xrightarrow{\supseteq} k_t$ (k_s is more general than k_t); $k_s \xrightarrow{\subseteq} k_t$ (k_s is less general than k_t); $k_s \xrightarrow{\equiv} k_t$ (k_s is equivalent to k_t); $k_s \xrightarrow{*} k_t$ (k_s is compatible with k_t); $k_s \xrightarrow{\perp} k_t$ (k_s is disjoint from k_t). Later in the paper we'll provide a formal definition of these five relations.

[5] Similar examples apply to catalogs. Here we use web directories, as they are well-known to most readers and easy to understand.

4 The algorithm: CTXMATCH

CTXMATCH takes the concept hierarchies of two HCs as input and returns a mapping between their nodes. The algorithm has the following two main steps:

Semantic explicitation: The meaning of each node k in a concept hierarchy H is made explicit in a logical formula $w(k)$. This formula approximates the intended meaning of the node k in H. For instance the formulae associated with the two nodes labeled `FLORENCE` on the right hand side of Figure 1 will approximate the meanings "images of Florence, a city in Tuscany" and "images of Florence, a city in Italy", respectively.

Semantic comparison: The problem of finding the semantic relation between two nodes $k \in H$ and $k' \in H'$ is encoded in a satisfiability problem, involving the formulae $w(k)$ and $w(k')$, and a background theory T containing properties (axioms) relevant for the relation between $w(k)$ and $w(k')$. So, to prove that the two nodes `FLORENCE` in Figure 1 are equivalent, we deduce the logical equivalence between the formulas associated to the nodes by using the domain axioms "Florence is a city of Tuscany" and "Tuscany is a region of Italy".

In the version of the algorithm presented here, we use WORDNET [12] as a source of both lexical and domain knowledge. However, WORDNET could be replaced by another combination of a linguistic resource and a domain knowledge resource.

4.1 Semantic explicitation

In this phase we make explicit the meaning of each node into a logical formula. Let us see how lexical, domain, and structural knowledge is exploited in this phase. Consider Figure 2. Using lexical knowledge, we associate linguistic senses to labels. For example, the label "Arizona" is associated with two senses corresponding to "a state in southwestern United States" or a "glossy snake". Domain knowledge and structural knowledge are used to filter out some of the possible senses.

Semantic explicitation is performed in two phases: *linguistic interpretation* and *contextualization*.

Linguistic interpretation In this first phase we provide an interpretation of the labels independently from the structure in which they occur. Let $H = \langle K, E, l \rangle$ be a concept hierarchy and L_H the set of labels associated to the nodes and edges of a hierarchy H by the function l. In this phase we associate to each label $s \in L_H$ a logical formula representing all possible linguistic interpretations of that label allowed by the lexical knowledge available.

Definition 4 (Label interpretation). *Given a logic W, a* label interpretation *in W is a function $li : L_H \rightarrow wff(W)$, where $wff(W)$ is a set of well formed formulas of W.*

The choice of W 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 have adopted the propositional fragment of description logic with $\sqcup$, $\sqcap$ and $\neg$, whose primitive concepts are the synsets of WORDNET that we associate to each label. Labels are processed by text chunking (via Alembic chunker [10]), and translation of the connectives into a logical form according to the following rules:

- coordinating conjunctions and commas are interpreted as a disjunction;
- prepositions, like 'in' or 'of', are interpreted as a conjunction;
- expressions denoting exclusion, like 'except' or 'but not', are interpreted as negations.

We access WORDNET to attach to each word in each label its set of senses. When two or more words in a label are contained in WORDNET as a single expression (a so-called multiword), the corresponding senses are selected and, in the basic logical form, the intersection between the two words is substituted by the multiword.

Example 1.

- $li(\texttt{Baroque}) = \texttt{baroque\#1}$, the unique sense of 'Baroque' presents in WORDNET;
- $li(\texttt{Arizona}) = \texttt{arizona\#1} \sqcup \texttt{arizona\#2}$, i.e., the disjunction of the two possible senses of 'Arizona';
- $li(\texttt{Chat and Forum}) = \texttt{chat\#1} \sqcup \texttt{chat\#2} \sqcup \texttt{chat\#3} \sqcup \texttt{forum\#1} \sqcup \texttt{forum\#2} \sqcup \texttt{forum\#3}$ i.e. the disjunction of the meaning of 'chat' and 'forum' taken separately (both 'chat' and 'forum' have tree senses in WORDNET);
- $li(\texttt{Classic Music}) = ((\texttt{classic\#1} \sqcup \ldots) \sqcap (\texttt{music\#1} \sqcup \ldots)) \sqcup \texttt{classic\_music\#1}$ either the conjunction of the meaning of 'classic' (with *n* senses) and the meaning of 'music' (with *m* sense) taken separately, or the multiword 'classic music' considered as a whole concept, ('classic music' is a multiword in WORDNET).

Contextualization The aim of this phase is to determine the component of the meaning of a node's label that depends on its position in the concept hierarchy associated to a HC. To this end, we introduce the notion of *focus* of a concept k in a hierarchy H, denoted by $f(k,H)$. Intuitively, the focus is the smallest sub-tree of H that one should take into account to determine the meaning of k in H. In CTXMATCH, the focus is defined as follows:

Definition 5 (Focus). *The* focus *of a node $k \in K$ in a concept hierarchy $H = \langle K, E, l \rangle$, is a finite concept hierarchy $f(k,H) = \langle K', E', l' \rangle$ such that: $K' \subseteq K$, and K' contains exactly k, its ancestors, and their children; $E' \subseteq E$ is the set of edges between the concepts of K'; l' is the restriction of l on K'.*

This definition of focus is motivated by observations on how we humans use HCs. When searching for documents in a HC, we incrementally construct the meaning of a node k by navigating the classification from the root to k. During this navigation, we have access to the labels of the ancestors of k, and also to the labels of their siblings. This information is used at each stage to select the node we want to access[6].

[6] This definition of focus is appropriate for HCs. With structures used for different purposes, different definitions of focus should be used. For example, if a concept hierarchy is used to

Given a focus $f(k,H)$ and the linguistic interpretation $li(.)$ of the labels of all its nodes, the phase called *contextualization* defines a formula $w(k)$ which is called the *structural interpretation* of the node k. We first set $w(k) := li(l(k))$ (i.e., $w(k)$ is the linguistic interpretation of the label associated to k), then we refine this definition via *sense filtering* and *sense composition*.

Sense filtering is a heuristic method by which we keep only the senses of a linguistic interpretation that a node k is more likely to have, and discharge the other ones. To this end, we analyze the relations between the senses of k and the senses of the other nodes in the focus. For example, if $w(k) = \texttt{arizona\#1} \sqcup \texttt{arizona\#2}$, the sense `arizona#2` (i.e., the snake) can be discharged if $f(k,H)$ contains the sense `United States#1` (the United States of America), and the focus does not contain any sense that is somehow related with snakes.

Sense composition enriches the meaning of a node's label by combining its linguistic interpretation with structural information and domain theory. For HCs, the rule is that the structural meaning of a concept k is formalized as the conjunction of the senses associated to all its ancestors; this makes sense, if we consider how we interpret the relation between a node and its ancestors in a classification. In CTXMATCH, some interesting exceptions are handled. For example, in the Yahoo!^{TM}Directory, `Visual arts` and `Photography` are sibling nodes under `Arts & Humanities`; since in WORDNET photography is in a is–a relationship with visual art, the node `Visual arts` is re-interpreted as visual arts with the exception of photography, and is then formalized in description logic as: $\texttt{visual art\#1} \sqcap \neg \texttt{photography\#1}$.

4.2 Computing semantic relations via SAT

After semantic explicitation is over, the problem of discovering semantic relations between two nodes k and k' in two HCs can be reduced to the problem of checking if a logical relation holds between the formulas $w(k)$ and $w(k')$; this is done again on the basis of domain knowledge. In CTXMATCH, the existence of a logical relation is checked as a problem of propositional satisfiability (SAT), and then computed via a standard SAT solver. The SAT problem is built in two steps. First, we select the portion B of the available domain knowledge which is relevant to the structural interpretation $w(k)$ and $w(k')$ of the two nodes k and k'; then we compute the logical relation between $w(k)$ and $w(k')$ which are implied by B.

Definition 6 (Background theory). *Let $\phi = w(k)$ and $\psi = w(k')$ be the structural interpretations of two nodes k and k' of two hierarchical classifications H_1 and H_2 respectively. Let T be a theory (a set of axioms) in the logic where ϕ and ψ are expressed. The* portion of T relevant to the semantic relation of ϕ and ψ, *denoted by $B(\phi,\psi)$ is a subset of T, such that for any subset B' of T, with $B(\phi,\psi) \subseteq B'$, we have that*

$$B' \models \alpha_{\phi,\psi} \;\; \textit{iff} \;\; B(\phi,\psi) \models \alpha_{\phi,\psi}$$

represents an XML-schema, the meaning of a node is determined also by the meaning of its sub-nodes, so a more suitable definition of focus $f(k,H)$ would include for example the subtree rooted at k.

where $\alpha_{\phi,\psi}$ is a formula obtained by combining ϕ and ψ by replacing all the atomic proposition of α either with ϕ or with ψ.

In the first version of CTXMATCH, the background theory B is built by transforming WORDNET relations between senses in a set of subsumption axioms as follows:

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

To build $B(\phi,\psi)$ from WORDNET, we adopt heuristic rules that turned out to produce satisfactory results. The idea is to extract the smallest set of axioms which provide semantic relations between senses that occur in ϕ and ψ. However, different sources (e.g., domain specific ontologies) and different heuristic rules may be used to build the background theory for ϕ and ψ.

Example 2. Suppose that we want to discover the relation between `Chat and Forum` in the Google™ Directory and `Chat and Forum` in the Yahoo!™Directory in Figure 2. From WORDNET we can extract the following relevant axioms: `art#1` $\sqsubseteq$ `humanities#1` (the sense 1 of 'art' is an hyponym of the sense 1 of 'humanities'), and `humanities#1` $\sqsupseteq$ `literature#2` (the sense 1 of 'humanities' is an hyperonym of the sense 2 of 'literature').

Once we have extracted a suitable background theory, we are ready to state a SAT problem for each possible relation in *rel* between any two nodes k and k' belonging to different HCs. In CTXMATCH, we use the following encoding:

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

B is the portion of the background theory relevant to k_s and k_t. The idea under this translation is to see WORDNET senses (contained in $w(k)$ and $w(k')$) as sets of documents. For instance the concept `art#i`, corresponding to the first WORDNET sense of art, is though as the set of documents speaking about art in the first sense. Using the set theoretic interpretation of mapping given in definition 7, we have that mapping can be translated in terms of subsumption of $w(k)$ and $w(k')$. Indeed subsumption relation semantically corresponds to the subset relation.

Example 3. The problem of checking whether `Chat and Forum` in Google™ is, say, less general than `Chat and Forum` in Yahoo!™ amounts to a problem of satisfiability on the following formulas:

ϕ = (art#1 $\sqcap$ literature#2 $\sqcap$ (chat#1 $\sqcup$ forum#1))
ψ = (art#1 $\sqcup$ humanities#1) $\sqcap$ humanities#1 $\sqcap$ (chat#1 $\sqcup$ forum#1)
$B(\phi,\psi)$ = (art#1 $\sqsubseteq$ humanities#1), (humanities#1 $\sqsupseteq$ literature#2)
It is easy to see that from the above axioms we can infer $B(\phi,\psi) \models \phi \sqsubseteq \psi$

5 Testing the algorithm

In this section, we report from [17] some results of the first tests on CTXMATCH. The tests were performed on real HCs (i.e., pre-existing classifications used in real applications), and not on *ad hoc* HCs.

In [1], a testing methodology is defined which is based on an ideal situation where two agents have the same set of documents and proceed to classify them into two different HCs following the *specificity principle* (see Section 3). Then, we can define the following criterion of correctness for mapping elements:

Definition 7 (Correctness of a mapping element[7]). *Let H_s to H_t be the concept hierarchies of two HCs. Let k_s and k_t denote any pair of nodes of H_s and H_t respectively. Let μ_s and μ_t denote two classifications of a set of documents D in H_s and H_t respectively. Then:*

1. $k_s \xrightarrow{\supseteq} k_t$ *is correct if for all* μ_s *and* μ_t, $\mu_s(k_s\downarrow) \supseteq \mu_t(k_t\downarrow)$;
2. $k_s \xrightarrow{\subseteq} k_t$ *is correct if for all* μ_s *and* μ_t, $\mu_s(k_s\downarrow) \subseteq \mu_t(k_t\downarrow)$;
3. $k_s \xrightarrow{\equiv} k_t$ *is correct if* $k_s \xrightarrow{\subseteq} k_t$ *is correct and* $k_s \xrightarrow{\supseteq} k_t$ *is correct;*
4. $k_s \xrightarrow{\perp} k_t$ *is correct if for all* μ_s *and* μ_t, $\mu_s(k_s\downarrow) \cap \mu_t(k_t\downarrow) = \emptyset$;
5. $k_s \xrightarrow{*} k_t$ *is correct if there is pair* μ_s *and* μ_t *such that* $\mu_s(k_s\downarrow) \cap \mu_t(k_t\downarrow) \neq \emptyset$.

where $\mu(c\downarrow)$ is the union of $\mu(d)$ for any d in the subtree rooted at c. A mapping is correct if all its elements are correct.

5.1 Experiment 1: Matching Google with Yahoo!

The aim of this experiment was to evaluate the CTXMATCH algorithm over portions of the Google™ Directory and the Yahoo!™ Directory about overlapping domains. The test was performed on the two sub-hierarchies 'Architecture' and 'Medicine' available in Google™ and Yahoo!™. The results, expressed in terms of precision and recall, are reported in the following table:

[7] The semantics introduced in Definition 7 can be viewed as an instance of the compatibility relation between contexts as defined in Local Models Semantics [13, 5]. Indeed, suppose we take a set of documents D as the domain of interpretation of the local models of two contexts c_1 and c_2, and each concept as a unary predicate. If we see the documents associated to a concept as the interpretation of a predicate in a local model, then the relation we discover between concepts of different contexts can be viewed as a compatibility constraint between the local models of the two concepts. For example, if the algorithm returns an equivalence between the concepts k_1 and k_2 in the contexts c_1 and c_2, then it can be interpreted as the following constraint: if a local model of c_1 associates a document d to k_1, then any compatible model of c_2 must associate d to k_2 (and vice versa); analogously for the other relations.

	Architecture		Medicine	
Relations	Pre.	Rec.	Pre.	Rec.
equivalence $\xrightarrow{\equiv}$	.75	.08	.88	.09
less general than $\xrightarrow{\subseteq}$	.84	.79	.86	.61
more general than $\xrightarrow{\supseteq}$	.94	.38	.97	.35

We observe that the use of domain knowledge allowed us to discover non trivial mappings. For example, an inclusion mapping was computed between `Architecture/History/Periods_and_Styles/Gothic/Gargoyles` and `Architecture/History/Medieval` as a consequence of the relation between `Medieval` and `Gothic` that can be found in WORDNET. This kind of semantic mappings are very difficult to find using a keyword–based approach.

5.2 Experiment 2: Product Re-classification

The second test was in the domain of e–commerce. In the framework of a collaboration with a worldwide telecommunication company, the matching algorithm was applied to re-classify the HC of the 'equipment and accessories' office (used to classify company suppliers) into UNSPSC[8] (version 5.0.2).

To evaluate the results of the re-classification, consider the different results between CTXMATCH and the baseline matching process[9]:

	Baseline classification		Matching classification	
Total items	194	100%	194	100%
Rightly classified	75	39%	134	70%
Wrongly classified	91	50%	16	8%
Non classified	27	14%	42	22%

Given the 194 items to be re-classified, the baseline process found 1945 possible nodes, but only 75 out of the 1945 proposed nodes are correct. The baseline, a simple string-based matching method, is able to capture a certain number of re-classifications, but the percentage of error is quite high (50%), with respect to the one of correctness (39%). Concerning the results of CTXMATCH, the percentage of success is significantly higher (70%) and, even more relevant, the percentage of error is minimal (8%).

6 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

[8] UNSPSC (Universal Standard Products and Services Classification) is an open global coding system that classifies products and services. UNSPSC is extensively used around the world for electronic catalogs, search engines, e–procurement applications and accounting systems.

[9] The baseline has been performed by a simple keyword based matching which worked according to the following rule: for each item description (made up of one or more words) gives back the set of nodes, and their paths, which maximize the occurrences of the item words.

	graph matching	CUPID	MOMIS	GLUE	CTXMATCH
Structural knowledge	•	•	•		•
Lexical knowledge		•	•	•	•
Domain knowledge				•	•
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 1. Comparing CTXMATCH with other methods

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 domain knowledge is used; (4) if instances are considered; (5) the type of result returned. The general results of our comparison are reported in Table 1.

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 and domain knowledge. Some examples of this approach are described in [21, 20, 19, 18, 14].

CUPID [16] 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`

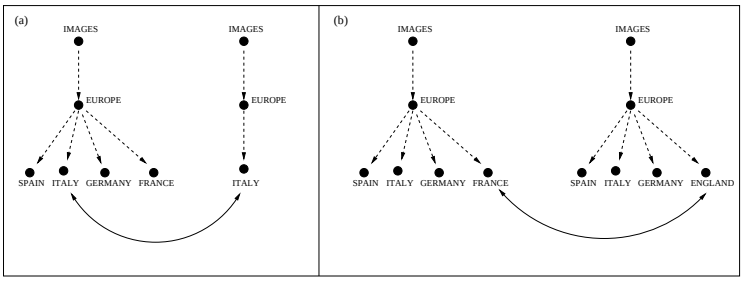

Fig. 3. Two example of mappings from CTXMATCH

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 allow an 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 [11] 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.

7 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. The algorithm has already been used in a peer-to-peer application for distributed knowledge management (the application is described in [3]), and is going to be applied in a peer-to-peer wireless system for ambient intelligence [9].

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 4) 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); 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. 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. P. Avesani. Evaluation framework for local ontologies interoperability. In *MeaN-02 – AAAI workshop on Meaning Negotiation*, Edmonton, Alberta, Canada, 2002.
2. Sonia Bergamaschi, Silvana Castano, and Maurizio Vincini. Semantic integration of semistructured and structured data sources. *SIGMOD Record*, 28(1):54–59, 1999.
3. M. Bonifacio, P. Bouquet, G. Mameli, and M. Nori. Kex: a peer-to-peer solution for distributed knowledge management. In D. Karagiannis and U. Reimer, editors, *Fourth Interna-*

tional Conference on Practical Aspects of Knowledge Management (PAKM-2002), Vienna (Austria), 2002.
4. M. Bonifacio, P. Bouquet, and P. Traverso. Enabling distributed knowledge management. managerial and technological implications. *Novatica and Informatik/Informatique*, III(1), 2002.
5. A. Borgida and L. Serafini. Distributed description logics: Directed domain correspondences in federated information sources. In R. Meersman and Z. Tari, editors, *On The Move to Meaningful Internet Systems 2002: CoopIS, Doa, and ODBase*, volume 2519 of *LNCS*, pages 36–53. Springer Verlag, 2002.
6. P. Bouquet, editor. *AAAI-02 Workshop on Meaning Negotiation*, Edmonton, Canada, July 2002. American Association for Artificial Intelligence (AAAI), AAAI Press.
7. G. C. Bowker and S. L. Star. *Sorting things out: classifi cation and its consequences*. MIT Press., 1999.
8. 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.
9. 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 Scientifica e Tecnologica (ITC-IRST), Trento (Italy), April 2003. Submitted to UbiComp'2003.
10. D. S. Day and M. B. Vilain. Phrase parsing with rule sequence processors: an application to the shared CoNLL task. In *Proc. of CoNLL-2000 and LLL-2000*, Lisbon, Portugal, September 2000.
11. 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.
12. Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. The MIT Press, Cambridge, US, 1998.
13. C. Ghidini and F. Giunchiglia. Local Models Semantics, or Contextual Reasoning = Locality + Compatibility. *Artifi cial Intelligence*, 127(2):221–259, April 2001.
14. Jeremy Carroll Hewlett-Packard. Matching rdf graphs. In *Proc. in the fi rst International Semantic Web Conference - ISWC 2002*, pages 5–15, 2002.
15. G. Lakoff. *Women, Fire, and Dangerous Things*. Chicago Unversity Press, 1987.
16. Jayant Madhavan, Philip A. Bernstein, and Erhard Rahm. Generic schema matching with cupid. In *The VLDB Journal*, pages 49–58, 2001.
17. 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.
18. 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.
19. Marcello Pelillo, Kaleem Siddiqi, and Steven W. Zucker. Matching hierarchical structures using association graphs. *Lecture Notes in Computer Science*, 1407:3–??, 1998.
20. 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.
21. 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-Verlag, Berlin.