ITC irst

CENTRO PER LA RICERCA
SCIENTIFICA E TECNOLOGICA

38050 Povo (Trento), Italy
Tel.: +39 0461 314312
Fax: +39 0461 302040
e−mail: prdoc@itc.it − url: http://www.itc.it

SEMANTIC COORDINATION IN SYSTEMS OF AUTONOMOUS AGENTS: THE APPROACH AND AN IMPLEMENTATION

Bouquet P., Serafini L., Zanobini S.

July 2003

Technical Report # 0307−07

Semantic coordination in systems of autonomous agents: the approach and an implementation

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 open agent-based systems. In this paper, we propose a new algorithm for discovering semantic mappings across autonomous agents' local data models based on a new approach to semantic coordination. This approach shifts the problem of semantic coordination from the problem of computing 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 why this is a significant improvement on previous approaches, and how it can be applied to a specific type of models, that is hierarchical classifications.

1 Introduction

One of the key issues in the development of open systems (e.g., the Semantic Web) is the problem of enabling agents 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 not viable for several reasons, such as the difficulty of "negotiating" a shared model of

[1] See the introduction of [4] for this notion, and its relation with the notion of *meaning negotiation*.

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

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* coordination, 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 4 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 new representation of a HC, in which all relevant knowledge about the nodes (including

[3] For an interesting discussion of the central role of classification in human cognition see, e.g., [10, 5].

their meaning in that specific HC) is encoded in 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 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.

2 Our approach

The approach to semantic coordination we propose in this paper is based on the intuition that there is a huge conceptual difference between coordinating generic abstract structures (e.g., arbitrary labelled graphs) and coordinating structures whose labels are expressions from the language spoken by the community of their users. Indeed, the latter ones 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. Interestingly enough, the status of this linguistic coordination is "codified" in artifacts (e.g., dictionaries, but today also ontologies and other formalized models), which provide senses for words, relations between their senses, and other knowledge about them. We want 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, human reasoners understand almost immediately that the relation between the first pair of nodes is "less general than" (after all, 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" (in fact, 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 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 map HCs?

The main elements of our approach can then be described as follows. 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 on the basis of 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

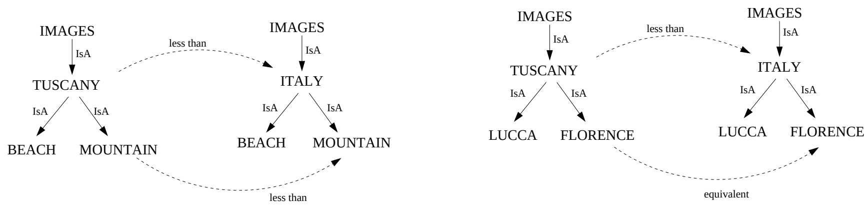

Fig. 1. Coordinating HCs

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 tells us 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 element 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 no longer stated as a matching problem, but can be encoded as a relatively simple problem of logical deduction. Intuitively, as we will say in a more technical form in Section 3, determining whether there is an equivalence relation between the meaning

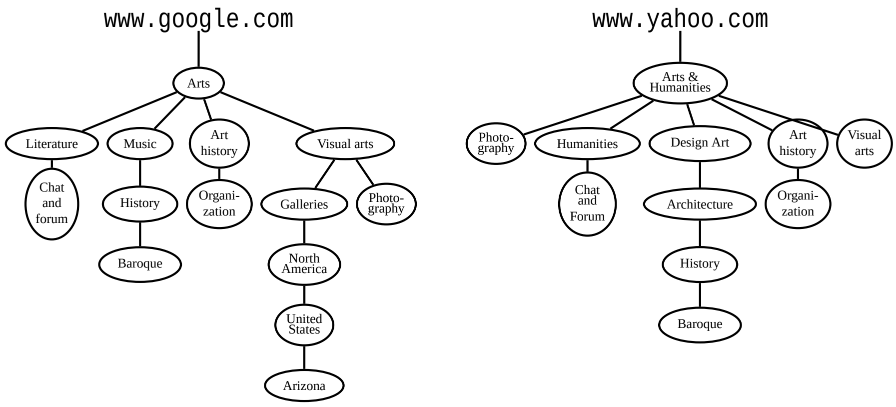

Fig. 2. Examples of hierarchical classification (source: Open Directory and Yahoo!Directory)

of two nodes becomes of problem of testing whether one implies the other and vice versa (given the available axioms); and determining whether one is less general than the other one amounts to testing if the first implies the second. 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 briefly an application of the general approach described in the previous section to the problem of coordinating HCs. See [13] for a more complete description. Intuitively, a classification is a grouping of documents into classes or categories. When categories are arranged into a hierarchical structure, we have a hierarchical classification. Prototypical examples of HCs are the web directories of many search engines, for example the Google^{TM} Directory, the Yahoo!^{TM} Directory, or the Looksmart^{TM} web directory. A tiny fraction of the HCs corresponding to the Google^{TM} Directory and to the Yahoo!^{TM}Directory is depicted in Figure 2.

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[4]. Formally, we define the problem of semantic coordination as the problem

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

of discovering mappings between categories in two distinct hierarchical classification H and H'.

The set of possible mappings depends on the intended use of the structures we want to map: Indeed, in our experience, this 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, the possible mappings include the following relations: $k_s \xrightarrow{\supseteq} k_t$, for k_s is more general than k_t; $k_s \xrightarrow{\subseteq} k_t$ for k_s is less general than k_t; $k_s \xrightarrow{*} k_t$ for k_s is compatible with k_t; $k_s \xrightarrow{\perp} k_t$ for k_s is disjoint from k_t; $k_s \xrightarrow{\equiv} k_t$ for k_s is equivalent to k_t. The meanings of these mappings are the obvious ones: $k_s \xrightarrow{\supseteq} k_t$ means that all documents classified in k_t could be classified also in k_s, but not vice-versa. Similar considerations for other mappings.

Formally, the CTXMATCH algorithm, developed for these purposes, takes as input two HCs and returns a set of mappings between their nodes. It is composed by two main steps: *Semantic explicitation* and *semantic comparison*[5]

Semantic explicitation: The meaning of each node k in a hierarchical classification H is made explicit in a logical formula $w(k)$, which approximates the intended meaning of the node k in H. The dimensions considered in this phase are three: lexical, domain, and structural knowledges. Consider the Figure 2. By *lexical knowledge* we can parse and associates 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* contribute to select some of these senses (the more appropriate in the context) and to build up the formula starting from them. For instance, from domain knowledge we have that Arizona is a state of United States, then we can discard the sense of "Arizona" as a glossy snake, retaining the sense of state, and we can conclude that the meaning of the node labelled with "Arizona" is exactly "the galleries of art in Arizona, the state of U.S.A.".

In this phase, a large amount of possibilities are handled: For example, in the Yahoo!TMDirectory, `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: `visual art#1`$\sqcup\neg$`photography#1`.

Semantic comparison The problem of finding the semantic relation between two nodes 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".

The existence of a logical relation is checked as a problem of propositional satisfiability (SAT), and then computed via a standard SAT solver. In the first version of

[5] For a more detailed description see [13].

CTXMATCH, the background theory 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.

Once we have extracted a suitable background theory, we are ready to state a SAT problem. In CTXMATCH, we use the following encoding:

relation	SAT Problem
$k_s \xrightarrow{\supseteq} k'_t$	$T \models w(k_t) \sqsubseteq w(k_s)$
$k_s \xrightarrow{\subseteq} k_t$	$T \models w(k_s) \sqsubseteq w(k_t)$
$k_s \xrightarrow{\perp} k_t$	$T \models w(k_s) \sqcap w(k_t) \sqsubseteq \perp$
$k_s \xrightarrow{\equiv} k_t$	$T \models w(k_t) \sqsubseteq w(k_s)$ and $T \models w(k_s) \sqsubseteq w(k_t)$
$k_s \xrightarrow{*} k_t$	$w(k_s) \sqcap w(k_t)$ is consistent in T

where T is the set of axioms extracted by the background knowledge.

For example, if we want to check whether `Chat and Forum` in Google^{TM} is, say, less general than `Chat and Forum` in Yahoo!^{TM}, we have to check if the formula

$$(\texttt{art\#1} \sqcap \texttt{literature\#2} \sqcap (\texttt{chat\#1} \sqcup \texttt{forum\#1})) \sqsubseteq$$
$$(\texttt{art\#1} \sqcup \texttt{humanities\#1}) \sqcap \texttt{humanities\#1} \sqcap (\texttt{chat\#1} \sqcup \texttt{forum\#1})$$

is satisfied by means of the following axioms:

$$\texttt{art\#1} \sqsubseteq \texttt{humanities\#1} \tag{1}$$
$$\texttt{humanities\#1} \sqsupseteq \texttt{literature\#2} \tag{2}$$

It is easy to see that it is true.

In the version of the algorithm presented here, we use WORDNET 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. Furthermore, in [12] are described some results of the first tests on CTXMATCH, performed on real HCs (i.e., pre-existing classifications used in real applications), and not on HCs created *ad hoc*.

4 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

	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.

A first family of approaches is based on graph matching techniques. Here, a HC is taken just 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 [17, 16, 15, 14, 9].

CUPID [11] 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 HCs 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

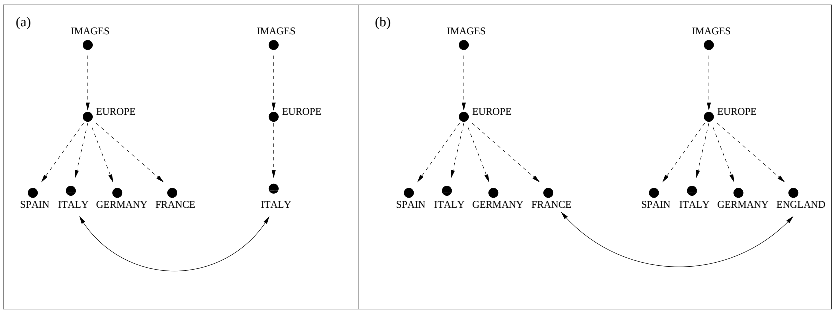

Fig. 3. Two example of mappings from CTXMATCH

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 the neighbor nodes match), then mapping 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) [1] 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 [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 many data are available, while CTXMATCH is more appealing 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 could not 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. The algorithm has already been used in a peer-to-peer application for distributed knowledge management (the application is described in [2]), and is going to be applied in a peer-to-peer wireless system for ambient intelligence [6].

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, HCs), but on the basis of the intended use of the structures under consideration. In this paper, we addressed the problem of coordinating HCs when used to build hierarchical classifications. Other possible uses of structures are: conceptualizing some domain (ontologies), describing a 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); 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 world knowledge.

References

1. Sonia Bergamaschi, Silvana Castano, and Maurizio Vincini. Semantic integration of semistructured and structured data sources. *SIGMOD Record*, 28(1):54–59, 1999.
2. 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 International Conference on Practical Aspects of Knowledge Management (PAKM-2002)*, Vienna (Austria), 2002.

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, editor. *AAAI-02 Workshop on Meaning Negotiation*, Edmonton, Canada, July 2002. American Association for Artificial Intelligence (AAAI), AAAI Press.
5. G. C. Bowker and S. L. Star. *Sorting things out: classification and its consequences*. MIT Press., 1999.
6. 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.
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. Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. The MIT Press, Cambridge, US, 1998.
9. Jeremy Carroll Hewlett-Packard. Matching rdf graphs. In *Proc. in the first International Semantic Web Conference - ISWC 2002*, pages 5–15, 2002.
10. G. Lakoff. *Women, Fire, and Dangerous Things*. Chicago Unversity Press, 1987.
11. Jayant Madhavan, Philip A. Bernstein, and Erhard Rahm. Generic schema matching with cupid. In *The VLDB Journal*, pages 49–58, 2001.
12. 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.
13. P. Bouquet B. Magnini, L. Serafini, and S. Zanobini. A SAT–based algorithm for context matching. In P. Blackburne, C. Ghidini, R.M. Turner, and F. Giunchiglia, editors, *Proceedings of the 4th International and Interdisciplinary COnference on Modeling and Using Context (CONTEXT-03). Stanford University (CA), June 23-25, 2003*, volume 2680 of *Lecture Notes in Artificial Intelligence*. Springer Verlag, 2003.
14. 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.
15. Marcello Pelillo, Kaleem Siddiqi, and Steven W. Zucker. Matching hierarchical structures using association graphs. *Lecture Notes in Computer Science*, 1407:3–??, 1998.
16. 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.
17. 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.