Asking and answering semantic queries*

P. Bouquet, G. Kuper, M. Scoz, S. Zanobini

Department of Information and Communication Technology – University of Trento
Via Sommarive, 10 – 38050 Trento (Italy)
{bouquet, kuper, scoz, zanobini}@dit.unitn.it

Abstract. One of the main issues in the development of the Semantic Web is the design and implementation of query languages that allow users to retrieve information from semantically annotated sources. In this paper, we describe a general methodology for querying a distributed collection of semantically heterogeneous resources, linked to each others through a collection of semantic mappings. The main contribution of this paper is the definition of *semantic query*, namely a query which enables users to tune a collection of semantic parameters to formulate the intended request. We show why this is different from what is typically done in data integration and peer-to-peer query reformulation.

1 Introduction

One of the main issues in the development of the Semantic Web is the design and implementation of query languages that allow users to retrieve information from semantically annotated sources. This problem, and several proposals have been put forward.

This problem has two fundamental dimensions. The first, which we call the *local dimension*, has to do with the problem of querying a single knowledge source (for example, an RDF [12] or an OWL [9] knowledge base) whose structure is known *a priori* and semantic heterogeneity is not a serious issue. A solution to this problem essentially amounts to proposing a query language (or family of languages) that does for Semantic Web languages what SQL does for relational databases. This problem neglects a crucial aspect of the Semantic Web, namely that in most real situations information will be distributed over a collection of distributed resources. This introduces the second dimension of the problem, called the *distributed dimension*, namely the problem of querying a collection of knowledge sources whose structure is not known *a priori* and where the degree of semantic heterogeneity can be quite high. Our work is focused on this second dimension. Relevant work in this area can be divided into two classes: global schema and peer-to-peer approaches. The first includes approaches based on some form of global schema. The idea is that a solution to the distributed dimensions of the querying problem requires the construction of a global schema which is then used to reformulate queries, either in a local as view (LAV) or global as view (GAV) architecture [13]. The second class includes peer-to-peer approaches, namely approaches in which the solution to the query problem is based on "horizontal" mappings across local

* The work presented in this paper was done as part of the EU funded project VIKEF (Virtual Information and Knowledge Environment Framework), contract n. 507173.

schemas. In such a scenario, a query can be thought of as a request formulated on a local schema to find semantically related data/information from a collection of remote schemas.

It was suggested (e.g. in [4]) that the problem of querying distributed and heterogeneous structures on a peer-to-peer basis can be divided into two main sub-problems: (i) the problem of discovering mappings across heterogeneous schemas (the *mapping problem*); and (ii) the problem of using a pre-existing collection of mappings to rewrite/ reformulate queries (the *query rewriting problem*). The idea is that first one needs to discover the (semantic) relation between two or more schemas; mappings are then used to answer queries over heterogeneous schemas, e.g. by reformulating a query written on a local schema into one or more queries on remote schemas.

In this paper, we argue that there is a further level, which we call the problem of *asking and answering semantic queries*. Indeed, the query rewriting problem can be restated as the problem of using semantic information (i.e., the available mappings) to reformulate "syntactic queries", namely queries that dig into the data associated to a schema by exploiting its structural properties (for XML-based languages, an example could be the rewriting of XPath expressions). But, in our view, a query is a semantic query only when the parameters used in its formulation are intrinsically semantic, namely are intended to refine the expression of a user's intended meaning. In other words, a semantic query is one whose result depends on parameters that are semantic in nature. Of course, an important question is what counts as a semantic parameter. In this paper we do not provide a general answer. However, since we will be mainly concerned with the problem of querying heterogeneous classifications, the relevant semantic parameters we will consider are: (i) the *type of relation*; (ii) the *ontological distance*; and (iii) the *lexical distance*.

2 The problem

Imagine that John is trying to find images for a book that he is writing about his holiday in Tuscany. Two web sites (say `PICS1` and `PICS2`) provide multi-media content. Figure 1 depicts a tiny portion of the structures they use to classify images. Suppose now that John is navigating the structure in `PICS1` and is interested in finding more images of Tuscany. John would like to ask something like: "Get me more documents which are related to what on this site is classified under `IMAGES/TUSCANY`".

Following what we said in the introduction, this request can be addressed at three different levels:

1. the first level corresponds to what in the introduction we called the mapping problem. It has to do with the discovery of the semantic relations between the categories of `PICS1` and `PICS2`. In Figure 1 we reported some possible mappings, for example that the category `IMAGES/TUSCANY` in `PICS1` is more specific than `PHOTOS/ITALY` in `PICS2`. At this level, one can say that there are many categories in `PICS2` semantically related to the category `IMAGES/TUSCANY` in `PICS1`, and that there are different possible relations (e.g. more general categories, or equivalent categories);

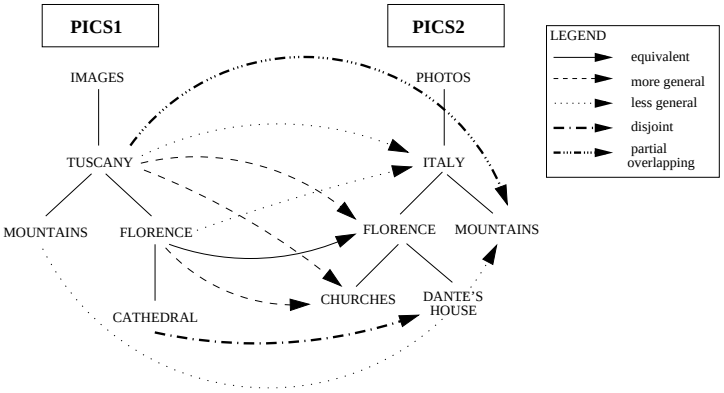

Fig. 1. Semantic mapping across classifications

2. the second level corresponds to the query rewriting problem. It refers to the fact that each semantic relation (i.e., each arrow from `IMAGES/TUSCANY` in `PICS1` to a category in `PICS2`) can be used to rewrite a query like `IMAGES/TUSCANY` on the schema `PICS1` into some query on the schema `PICS2`;
3. the third level corresponds to what we called the semantic query problem. It has to do with the fact that a query like `IMAGES/TUSCANY` on the schema `PICS1` does not provide enough information on what John may have in mind. For example: is he interested only in images which are classified under nodes that are semantically equivalent to the node `TUSCANY`? Is he willing to accept also images from more specific categories? If so, to what extent? Is a photo of Florence acceptable? And a photo of Dante's house in Florence? Adding this information to a standard query (or to any reformulation of a query) is what we define as asking a semantic query, and is the main focus of this paper.

The problem of allowing semantic queries can be divided into two sub-problems, which we call the *How-To-Ask* and the *How-To-Answer* problems respectively.

How-To-ask. The first class of problems is related to the parameters that John should be able to "tune" to specify his request. The parameters we are interested in are semantic parameters, namely parameters that can be used to refine the interpretation of John's request. The three parameters we take into account in this paper are: (i) *type of relation*, which is used to restrict the query to categories which are in a specific semantic relation with the category of the source schema; (ii) the *ontological distance*, which is used to specify the acceptable distance from the category in the source schema and categories in other schemas (with respect to some reference ontology); and (iii) the *lexical distance*, which is used to tune the distance between the linguistic formulation of the category in the source schema and the linguistic formulation of categories in other schemas.

How-To-answer. Once a semantic query is formulated, there is the problem of answering it appropriately. In this paper, we will ignore the details of solving the "structural" part of the query, instead, we will focus on the semantic part, namely the resolution of

semantic constraints specified by users. To do this, we will assume that a collection of semantic relations across the different structures has already been computed by some matching algorithm[1], and show how these mappings can be used to answer a query which specifies the values of these semantic parameters. The "How-To-Answer" part of the problem is non-trivial, as it requires one to take into account the fact that the evaluation of semantic parameters depends on what knowledge is used. For example, two concepts that are ontologically very close for John might be very distant for Mary, in particular if they use different background ontologies to evaluate such a distance. Therefore we need to provide a solution in which it is clear whose knowledge is being used.

Returning to the example, John can ask the following semantic query: "Get me documents classified in categories which are equivalent or more specific than the category `IMAGES/TUSCANY` in `PICS1`, where the ontological distance is less or equal to n and the linguistic distance is unbounded". Intuitively, if we assume that at `PICS2` an ontology is available according to which Dante's house is in Florence, and Florence is in Tuscany, then we can deduce that the node `FLORENCE` in `PICS2` is ontologically less distant from the node `IMAGES/TUSCANY` in `PICS1` than the node `DANTE'S HOUSE`, even though they are both related to the source category by the same semantic relation (i.e. less general).

3 Semantic queries over distributed classifications

Considering a collection $S_1, \ldots, S_n$ of semantically heterogeneous structures and a set of mappings $\mathcal{M}_1, \ldots, \mathcal{M}_j$ across them, a distributed query is a request, posed on one of the structures, to retrieve data from the other structures. Such a query is a semantic query when the answer is based on the satisfaction of a collection of semantic parameters.

In this section we focus in particular on a specific scenario, where $S_1, \ldots, S_n$ are hierarchical classifications, such as Web directories or catalogs. For such an application, we now propose a precise notion of semantic query, in which semantic parameters are explicitly listed, and then define the notion of semantically appropriate answer. In the following section, we show that this notion of semantic query can be easily implemented on top of CTXMATCH, an algorithm presented in [3] which automatically generates mappings across hierarchical classifications.

3.1 Choosing the mapping

As stated in the introduction, the problem of semantic queries is different from the problem of discovering mappings across structures, and that semantic queries use pre-existing mappings. We now discuss what types of mappings are needed to support semantic queries.

Generally speaking, a mapping between two schemas S_1 and S_2 (including taxonomies, ontologies, catalogs) can be thought of as a triple $\langle n_1, n_2, R \rangle$, where n_1 is

[1] Section 4 describes one possible method for computing these relations, based on the work presented in [3].

a node of S_1, n_2 is a node of S_2, and R is a relation between the two nodes. These mappings are calculated in many different ways and the existing approaches can be classified in two main categories, depending on the nature of the relation they compute:

- methods that return numerical values (typically between 0 and 1), whose intended meaning is the semantic proximity between the two nodes. Examples include CUPID [14], MOMIS [1], and GLUE [6];
- methods that return semantic relations, i.e., a relation with a clear model-theoretic interpretation (e.g., logical equivalence or subsumption). Examples include CTX-MATCH [3], S-MATCH [10].

From the perspective of semantic queries, the problem with the first category is that the interpretation of the result is unclear. In fact, for example, could be difficult to interpret in the right way a 0.9 similarity? These questions are important if we want to allow such logical relation between concepts as semantic parameters. For this reason, we assume in this paper that the available mappings are a collection of coordination rules, defined as follows.

Definition 1 (Coordination rule). *A* coordination rule *from a structure $\mathcal{S}_A$ to a $\mathcal{S}_B$ is a quadruple $\langle id, m, n, r \rangle$, where:*

- *id is a unique identifier for the rule;*
- *$m \in \mathcal{S}_A$ and $n \in \mathcal{S}_B$ are nodes in the corresponding structures;*
- *r is the semantic relation holding between m and n.*

In [3], it is argued that, when the structures are classifications, the following set $\Re$ of semantic relations must be considered: $\equiv$ (equivalence), $\subset$ (the first is strictly less general than the second), $\supset$ (the first is strictly more general than the second), $*$ (partial overlapping), $\perp$ (exclusion). Relations are interpreted in terms of documents that would be classified under the two categories. Given a collection of documents D, $\equiv$ means that the same subset of D would be classified under the two categories, $\subset$ means that all documents classified under the category in the source structure would be classified also under the category of the target structure (and similarly for $\supset$), $*$ means that there is a possible intersection between the sets of documents classified under the two categories, $\perp$ means that no document can be classified under both categories.

A mapping is defined as a set of coordination rules:

Definition 2 (Mapping). *A* mapping *$\mathcal{M}_{A \to B}$ between two structures $\mathcal{S}_A$ and $\mathcal{S}_B$ is a pair $\langle id, \mathcal{CR} \rangle$, where id is a unique identifier for the mapping and $\mathcal{CR}$ is a set of coordination rules from nodes of $\mathcal{S}_A$ to nodes of $\mathcal{S}_B$.*

3.2 Choosing the relevant semantic parameters

We now discuss the types of parameters we consider. They may depend on the types of structures that are queried, or on the specific application. We propose a list of semantic parameters that, in our opinion, are among the most important for the specification of semantic queries.

In [3], it is argued that computing semantic mappings across hierarchical classifications depends on three different types of knowledge:

Ontological Knowledge. Ontological Knowledge ($\mathcal{O}$) represents what is known about a given domain, or about the world in general.[2]. Intuitively, $\mathcal{O}$ can be thought of as the set of 'objects', or concepts, that an agent has knowledge about together with some relations among them. Facts in the $\mathcal{O}$ used in our example include the fact that Florence is located in Tuscany, that Tuscany is part of Italy, that Italy is in Europe, and that Europe is a continent.

Lexical knowledge. Lexical knowledge represents knowledge about the relationship between the concepts of an ontology $\mathcal{O}$ and their encoding into the language that is d used to communicate with other agents. One of the best-known instances of lexical knowledge is WORDNET [7], but note that WORDNET also includes part of what we call ontological knowledge.

Structural knowledge. Structural knowledge refers to the fact that a classification typically classifies documents under categories that correspond to concepts which are not directly defined in the ontology, but are obtained from the "composition" of concepts defined in one or more ontologies. For example, the category 'photos of Italian mountains' from the schema `PICS2` in Figure 1 is obtained by combining the concepts photo, Italy, and mountains. This knowledge is called structural, as it is used to build the structure of the classification.

If we assume that a mapping is used as a way of rewriting queries for different schemas, then the parameters associated to a semantic query should be used to filtering out some of these rewritings. To make this possible, we introduce parameters that are related to the way that each of the three kinds of knowledge described above are used to compute each coordination rule in a mapping. The result is the following list of parameters:

1. Ontological distance. This parameter encodes the 'ontological effort' that is required for determining the semantic relation between two concepts. As an example, we show that the ontological distance between `PHOTOS/ITALY/FLORENCE` in `PICS2` and `IMAGES/TUSCANY` in `PICS1` is smaller that the ontological distance between `PHOTOS/ITALY/FLORENCE/DANTE'S HOUSE` in `PICS2` and `IMAGES/TUSCANY` in `PICS1`. Both pairs of nodes are connected via two coordination rules which contain the same relation ($\supset$). However, the computation of the second rule requires the use of more ontological knowledge, as it depends on at least two facts: that Florence is in Tuscany, and that Dante's house is in Florence. The computation of the first rule depends only on the first fact. This observation can be used to conclude that, given an ontology which contains these two facts, the derivation of the second coordination rule requires a greater "ontological effort" and therefore that the ontological distance is higher[3].

2. Lexical distance. This parameter represents the 'lexical effort' needed to determine if two words denote the same concept. The prototypical example of lexical effort is the substitution of a word with a synonym. However, other (not strictly semantic) techniques can be used to force words occurring in different schemas to refer to the same

[2] Here, we use the word 'ontology' in the broad sense of an explicit and formal conceptualization of the world. Indeed, at this level, we do not need to distinguish between types of ontologies, e.g. top level ontologies, domain ontologies, application ontologies, etc.

[3] We would like to stress the fact that the ontological distance does not express a structural distance between nodes, but only refers to how far a relation is from another w.r.t. the ontology.

concept. These include string manipulation, lemmatizers, *ad hoc* thesauri, etc. Lexical distance allows us to say, for example, that the concept `IMAGES/ITALY` is closer to `IMAGES/TUSCANY` in `PICS1` than to the concept `PHOTOS/ITALY` in `PICS2`. Indeed, even though one may argue that `IMAGES/ITALY` and `PHOTOS/ITALY` are the same concept, the computation of the coordination rule which determines their semantic relation with `IMAGES/TUSCANY` in `PICS1` requires a greater lexical effort, namely the use of the piece of lexical knowledge saying that, at least in one possible sense, the word 'PHOTO' and the work 'IMAGES' are synonyms. In this paper, we shall only consider only synonymy, in which case the lexical distance is a Boolean parameter with values 0 or 1. In general, however, it could be used as a real distance, with sophisticated techniques to introduce finer grained measures, where the similarity between words could be expressed as a real number between 0 and 1.

3. Type of relation. Each coordination rule represents a semantic relation between two complex concepts, i.e., concepts that are built from simple concepts defined in some ontology and organized in a classification structure. As stated above, there is may be more than one possible relation between two such concepts. This parameter is used to select the relation of interest for a given query. For example, it allows John to say that he wants only images that are classified under categories that are equivalent to the category `IMAGES/TUSCANY` in `PICS1`; in our example this would return the empty set.

3.3 Semantic queries

We can now define formally the notion of a semantic query.

Definition 3 (Semantic Query). *A semantic query $\mathcal{Q}$ is a 5-tuple $\langle \mathcal{S}, m, r_{\mathcal{M}}, \Delta_o, \Delta_l \rangle$, where:*

- *$\mathcal{S}$ is a structure;*
- *m is a node in $\mathcal{S}$;*
- *$r_{\mathcal{M}} \in \Re$ is a semantic relation;*
- *Δ_o is the ontological distance;*
- *Δ_l is the lexical distance;*

A *semantically appropriate answer* to a semantic query $\mathcal{Q}$ can be defined as follows:

Definition 4 (Semantically Appropriate Answer). *Let $\mathcal{M}$ be a mapping between a source structure $\mathcal{S}_A$ and a target structure $\mathcal{S}_B$, and let $\mathcal{Q}$ be a query. The* semantically appropriate answer *to $\mathcal{Q}$ is the set of nodes $n \in \mathcal{S}_B$ such that n is related to m through the mapping $r_{\mathcal{M}}$, i.e., $\langle id, m, n, r_{\mathcal{M}} \rangle \in \mathcal{M}$, for some values of id. Furthermore, n must be at the appropriate ontological and lexical distance from m.*

4 An example

We illustrate our general framework by showing how a semantic query engine can implemented on top of CTXMATCH, the algorithm for discovering semantic mappings across heterogeneous structures described in [3].

The input to CTXMATCH consists of two structures, and the result is a mapping between them. This mapping is computed in two main steps: (i) semantic explicitation, in which the meaning of each node of the two structures is made explicit and is encoded as a set of logical formulas; and (ii) semantic comparison, in which the problem of discovering the semantic relation between two nodes is now encoded as a relatively simple problem of logical deduction. Then, determining whether there is an equivalence relation between two nodes becomes a problem of testing whether the formulas associated to two nodes are logically equivalent, w.r.t. the appropriate axioms. Consequently it's used a standard SAT solver to checks the relations. Table 1 summarizes the satisfiability problems associated to each relation, where ϕ represents the meaning associated to a node in the source structure and ψ represents the meaning associated to a node in the target structure. The tests are performed in the order listed in this table, and the relation that is returned corresponds to the first positive answer[4].

	TEST	RELATION RETURNED
1	$\Theta \models \neg(\phi \wedge \psi)$	$\perp$
2	$\Theta \models \phi \equiv \psi$	$\equiv$
3	$\Theta \models \phi \rightarrow \psi$	$\subset$
4	$\Theta \models \psi \rightarrow \phi$	$\supset$
5	default	$*$

Table 1. Set of SAT tests

In the current version of CTXMATCH, ontological knowledge $\mathcal{O}$ is represented as a directed acyclic graph where nodes represent concepts and arcs represent roles.

Definition 5 (Ontological Knowledge). *Let C be a set of concepts, and R a set of roles.* Ontological knowledge *(denoted by $\mathcal{O}$) is a quadruple $\langle N, E, l, l' \rangle$ where N is a finite set of nodes, $E \subseteq N \times N$ is the set of arcs on N, $l : N \rightarrow C$ is a bijective function from the set N of nodes to the set C of concepts and $l' : E \rightarrow R$ is a function from the set E of arcs to the set R of roles.*

Lexical knowledge is a function which assigns sets of concepts to each word of a lexicon[5], where a lexicon is the set of words that are used to describe the concepts. Formally, let C be a set of concepts and L a set of lemmas. Then lexical knowledge is defined as follows:

[4] Note that the relation returned by tests 3 and 4 is strict containment, since the system performs test 3 only if a negative result was returned by test 2. If a positive answer is returned by 3, it means that $\Theta \models \phi \rightarrow \psi \wedge \neg(\phi \equiv \psi)$, which corresponds to the '$\subset$' relation. A similar explanation applies to the $*$ relation, which the default case.

[5] We allow sets of concepts in the lexical function since in most human languages the same lemma can express more that one concept (polysemy). In an ideal language, where no polysemy is possible, the sets L and C would be isomorphic and the lexicon function would be a bijective function.

Definition 6 (Lexical knowledge). Lexical knowledge *is a function* $\mathcal{L} : L \rightarrow 2^C$ *from lemmas to sets of concepts.*

The coordination rules returned by CTXMATCH already contain information about the semantic relations between pairs of nodes of two classifications. However, to implement the mechanism of semantic query on top of CTXMATCH, we also need to compute the lexical and ontological distance associated to each rule. In this paper we decide to precompute the values when creating the mapping, adding this information to the existing coordination rules [6].

The modified version of CTXMATCH therefore returns *extended coordination rules*:

Definition 7 (Extended Coordination Rule). *An* Extended Coordination Rule *from a structure* $\mathcal{S}_A$ *to a structure* $\mathcal{S}_B$ *is a 6-tuple* $\langle id, m, n, r, d, ld \rangle$*, where:*

- id*,* $m \in \mathcal{S}_A$*,* $n \in \mathcal{S}_B$ *and* r *are as in Definition 1;*
- d *is the ontological distance of the coordination rule;*
- ld *is the lexical distance of the coordination rule.*

Accordingly, we extend the definition of mapping as follows:

Definition 8 (Extended Mapping). *A* extended mapping $\mathcal{M}_{A \rightarrow B}$ *between two structures* $\mathcal{S}_A$ *and* $\mathcal{S}_B$ *is a set of Extended Coordination Rules.*

We now discuss how CTXMATCH actually computes the ontological and lexical distance above.

Let $\mathcal{O}$ be an ontology as defined in Definition 5, and let c and c' be two concepts in $\mathcal{O}$. We say that two concepts c and c' are *related* iff there is at least one path on the graph that connects the corresponding nodes $l^{-1}(c)$ and $l^{-1}(c')$. The ontological distance between c and c' is then defined as follows.

Definition 9 (Ontological Distance between simple concepts). *The* Ontological Distance *between* c *and* c'*, written* $D_s(c, c')$*, is the length of the minimal path between the nodes corresponding to* c *and* c' *in* $\mathcal{O}$*, if such a path exists, and is* 0 *otherwise.*

For example, if 'Florence $\xrightarrow{\text{Part-Of}}$ Tuscany $\xrightarrow{\text{Part-Of}}$ Italy' is the minimal path in $\mathcal{O}$ between the simple concepts 'Italy' and 'Florence', then the ontological distance D_s(Italy, Florence) is 2 (two arcs).

However, in general, we are interested in calculating the distance between two complex concepts. To define this distance, we introduce the following definitions.

Definition 10 (Ontological Distance between sets of simple concepts). *Let* A *and* B *be two sets of simple concepts. The* ontological distance *between the sets* A *and* B*,* $D_c(A, B)$*, is*

$$\Sigma_{c \in A,\, c' \in B} D(c, c')$$

[6] This fact does not increase the complexity of CTXMATCH.

The ontological distance between sets of simple concepts is the sum of the ontological distances of all the possible pairs of simple concepts in the two sets.

This definition involves some redundancy, and we therefore introduce the notion of normalized set of simple concepts.

Definition 11 (Normalized set of simple concepts). *Let K be the set of simple concepts occurring in a complex concept α. A* normalized set of simple concepts $K' \subseteq K = \{c \in K \mid$ *there is no path from c' to c in $\mathcal{O}$ for some $c' \in K\}$.*

For example, $K = \{\mathrm{Photos}, \mathrm{Italy}, \mathrm{Florence}\}$ is the set of simple concepts associated to the complex concept 'Images of Florence in Italy'. Then the normalized set K' is $\{\mathrm{Images}, \mathrm{Florence}\}$, as the presence of the *Part-Of* relation between 'Florence' and 'Italy' in the ontology $\mathcal{O}$ allows us to delete 'Italy' from the set.

The ontological distance between complex concepts can now be defined as follows.

Definition 12 (Ontological Distance between complex concepts). *Let A and B be the set of simple concepts occurring in complex concepts ϕ and ψ respectively. The* Ontological Distance between the complex concepts *ϕ and ψ is defined as $D_c(A', B')$, where A' and B' are the normalized sets of simple concepts for A and B respectively.*

For lexical distance, we introduce the notion of translation clause as follows.

Definition 13 (Translation clause). *Let k be a node in a structure $\mathcal{S}$ and ϕ the associated complex concept. Then a* translation clause *C_j for ϕ is a set of pairs $\langle w, \mathcal{L}(w)\rangle$, where w is a word occurring in one of the labels of the nodes lying in the path from root to k, and $\mathcal{L}(w) = \langle s_1, \ldots, s_n\rangle$ is the set of possible concepts denoted by the word w w.r.t. a lexicon $\mathcal{L}$.*

Consider the node TUSCANY of the right structure depicted in Figure 1. The translation clause for this node w.r.t. WORDNET (used as lexicon) is the set

$$\{\langle image, \langle image\#1, \ldots, image\#7\rangle\rangle, \langle Tuscany, \langle tuscany\#1\rangle\rangle\}$$

Definition 14 (Lexical Distance between complex concepts). *Let ϕ and ψ complex concepts and C_s and C_t be the 'translation clauses' for ϕ and ψ respectively. The* Lexical Distance *between ϕ and ψ is 1 if $C_t \subseteq C_s$, and is 0 otherwise.*

In this framework, the definition of *semantically appropriate answer* can be restated as follows:

Definition 15 (Semantically Appropriate Answer Specialized). *Let $\mathcal{M}$ be an extended mapping between a source structure $\mathcal{S}_A$ and a target structure $\mathcal{S}_B$, and let $\mathcal{Q}$ be a query. The* semantically appropriate answer *to $\mathcal{Q}$ is the set of nodes $n \in \mathcal{S}_B$ such that n is related to m through the mapping $r_{\mathcal{M}}$, i.e., $\langle id, m, n, r_{\mathcal{M}}, d, ld\rangle \in \mathcal{M}$, for some values of id. Furthermore, n is at the appropriate lexical distance, i.e. $ld \leq \Delta_l$, and ontological distance, i.e. $d \leq \Delta_o$, from m.*

We illustrate this definition with a simple example. We use a syntax is based on XPath, extended to allow the specification of semantic parameters. The notation for the semantic parameters is similar to XPath qualifiers, but using angle brackets to make the distinction clear. We can therefore qualify a node by using: (i) $\langle rel = r \rangle$, where $r \in \Re$ (i.e., $\equiv$, $\subset$, etc.); (ii) $\langle od \leq k \rangle$, to restrict the ontological distance to be less that of equal to k (strict equality can also be used); (iii) $\langle ld = 0 \rangle$ to restrict the lexical distance ($ld = 1$ could be used, but would be redundant).

Consider Figure 1. The query '$/\texttt{IMAGES}/\texttt{TUSCANY}\langle r_{\mathcal{M}} = \supset \rangle \langle \Delta_o \leq 1 \rangle \langle \Delta_l = 0 \rangle'$, expressed using a XPath expression on the source HC (on the left), specifies that the user wants documents that are contained in nodes that are semantically related to the path `IMAGES/TUSCANY` by means of a semantic relation $\supset$, but that are ontologically distant up to 1, and whose concepts are lexically equivalent.

Considering the mapping, we know that the path `IMAGES/TUSCANY` is related, in some way, to six elements. The first constraint is represented by the restriction on the semantic relation '$\supset$'. We can see that only the paths `PHOTOS/ITALY/FLORENCE/CHURCHES` and `PHOTOS/ITALY/FLORENCE/DANTE'S HOUSE` satisfy this constraint. The second constraint is that on the ontological distance, which is satisfied only be `PHOTOS/ITALY/FLORENCE/CHURCHES` respects the constraint. Indeed, the path `PHOTOS/ITALY/FLORENCE/CHURCHES` is at ontological distance 2 from `IMAGES/TUSCANY`, since 'Dante's house *Part-Of* Florence', and 'Florence *Part-Of* Tuscany'. The third constraint is that on the lexical distance. This is satisfied by none of the paths, as, the lexical distance between `IMAGES/TUSCANY` and `PHOTOS/ITALY/FLORENCE/CHURCHES` is 1, since the translation clause $\langle image, \langle s_1, \ldots, s_n \rangle \rangle$ is in C_t but not in C_s.

5 Related Work

We present here roadmap to relevant work in the area of querying the Semantic Web.

We first distinguish approaches that address the problem of querying a single knowledge source from those that address the issue of distributing queries across multiple (heterogeneous) sources. In the first group we find RQL [12, 11], XML-QL [5], and XQuery [2]). We also include DQL [8] and OWL-QL [9], even though they are designed for a distributed environment. Indeed, they address the problem of a client-server interaction, but, to the best of our knowledge, not the problem of querying semantically heterogeneous resources.

In the other group, there are two main approaches, those that use a global schema (GAV and LAV), and and P2P approaches. Among the P2P approaches, we classify the relevant work according to the three levels discussed in the introduction: the mapping level, the query rewriting level, and the semantic query level.

We have explained why the mapping problem is different from the problem of using mappings to answer queries. The approach that is closest to ours is that of [4]. In this paper, the two levels are (i) a particular kind of mapping between the exported fragments V_l and V_r of a knowledge base from the local and the remote peers, and (ii) the query problem, regarded as the problem of rewriting a query on the local structure into another query on the remote structure using the ontological knowledge encoded in the mapping.

These two levels are different from our *mapping* and the *semantic query* levels, and both of the problems of [4] are addressed at the mapping level. A set of semantic relations relating a path m in a structure $\mathcal{S}_A$ with a set of paths $n_1, \ldots, n_k$ in another structure $\mathcal{S}_B$ represents the set of all the possible rewritings in $\mathcal{S}_B$ of a query q on m, so that the 'syntactical rewriting' of the query q would be redundant. Furthermore, our semantic query level adds a further level called the *asking and answering problem*.

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. S. Boag, D. Chamberlin, M. Fernandez, D. Florescu, J. Robie, and J. Simeon. XQuery 1.0: An XML query language. Technical report, W3C, November 2003.
3. 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, Sanibel Island (Florida, USA), October 2003.
4. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. What to ask to a peer: ontology-based query reformulation. In *9th International Conference on Principles of Knowledge Representation and Reasoning (KR-2004)*, 2004.
5. A. Deutsch, M. Fernandez, A. Levy, and D. Suciu. XML-QL: A query language for XML. Technical report, W3C, August 1998.
6. A. Doan, J. Madhavan, P. Domingos, and A. Halevy. Learning to map between ontologies on the semantic web. In *11th Int. WWW Conf., Hawaii*, 2002.
7. Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. The MIT Press, Cambridge, US, 1998.
8. R. Fikes, P. Hayes, and I. Horrocks. DQL - a query language for the semantic web. Technical report, Knowledge Systems Laboratory, 2002.
9. R. Fikes, P. Hayes, and I. Horrocks. OWL-QL - a language for deductive query answering on the semantic web. Technical report, Knowledge Systems Laboratory, Stanford, CA, 2003.
10. F. Giunchiglia and P. Shvaiko. Semantic matching. *Proceedings of the workshop on Semantic Integration*, October 2003.
11. G. Karvounarakis. The RDF query language (RQL). Technical report, Institute of Computer Science, Foundation of Research Technology, 2003.
12. G. Karvounarakis, S. Alexaki, V. Christophides, D. Plexousakis, and M. Scholl. RQL: A declarative query language for RDF. In *Proc. of the eleventh international world wide web conference*, Honolulu, Hawaii, USA, May 2002.
13. A. Calì, D. Calvanese, G. De Giacomo, and M. Lenzerini, Data integration under integrity constraints, In *Information Systems*, pages 147–163, 2004.
14. Jayant Madhavan, Philip A. Bernstein, and Erhard Rahm. Generic schema matching with cupid. In *The VLDB Journal*, pages 49–58, 2001.