Ostensive Automatic Schema Mapping for Taxonomy-based Peer-to-Peer Systems

Yannis Tzitzikas[1] and Carlo Meghini

Istituto di Scienza e Tecnologie dell' Informazione [ISTI]
Consiglio Nazionale delle Ricerche [CNR], Pisa, Italy
Email : {tzitzik|meghini}@iei.pi.cnr.it

Abstract This paper considers Peer-to-Peer systems in which peers employ taxonomies for describing the contents of their objects and for formulating semantic-based queries to the other peers of the system. As each peer can use its own taxonomy, peers are equipped with inter-taxonomy mappings in order to carry out the required translation tasks. As these systems are ad-hoc, the peers should be able to create or revise these mappings on demand and at run-time. For this reason, we introduce an ostensive data-driven method for automatic mapping and specialize it for the case of taxonomies.

1 Introduction

There is a growing research interest on peer-to-peer systems like Napster, Gnutella, FreeNet and many others. A peer-to-peer (P2P) system is a distributed system in which participants (the peers) rely on one another for service, rather than solely relying on dedicated and often centralized servers. Many examples of P2P systems have emerged recently, most of which are wide-area, large-scale systems that provide content sharing [4], storage services [19], or distributed "grid" computation [2, 1]. Smaller-scale P2P systems also exist, such as federated, server-less file systems [10, 7] and collaborative workgroup tools [3].

Existing peer-to-peer (P2P) systems have focused on specific application domains (e.g. music file sharing) or on providing file-system-like capabilities. These systems do not yet provide semantic-based retrieval services. In most of the cases, the name of the object (e.g. the title of a music file) is the only means for describing the contents of the object. Semantic-based retrieval in P2P systems is a great challenge. In general, the language that can be used for indexing the objects of the domain and for formulating semantic-based queries, can be *free* (e.g natural language) or *controlled*, i.e. object descriptions and queries may have to conform to a specific vocabulary and syntax. The first case, resembles distributed Information Retrieval (IR) systems and this approach is applicable in the case where the objects of the domain have a textual content (e.g. see

[1] Work done during the postdoctoral studies of the author at CNR-ISTI as an ERCIM fellow.

[23]). In this paper we focus on the second case where the objects of a peer are indexed according to a specific conceptual model represented in a data model (e.g. relational, object-oriented, logic-based, etc), and content searches are formulated using a specific query language. This approach, which can be called "database approach", starts to receive noteworthy attention by the researchers, as is believed that the database and knowledge base research has much to contribute to the P2P grand challenge through its wealth of techniques for sophisticated semantics-based data models and query processing techniques (e.g. see [14, 9, 18, 15, 32]). A P2P system might impose a single conceptual model on all participants to enforce uniform, global access, but this will be too restrictive. Alternatively, a limited number of conceptual models may be allowed, so that traditional information mediation and integration techniques will likely apply (with the restriction that there is no central authority). The case of fully heterogeneous conceptual models makes uniform global access extremely challenging and this is the case that we are interested in.

The first and basic question that we have to investigate is which conceptual modeling approach is appropriate for the P2P paradigm. We would like a scalable conceptual modeling approach which also allows bridging the various kinds of heterogeneity in a systematic and easy manner. As there are no central servers, or mediators, each participating source must have (or be able to create) *mappings*, or articulations, between its conceptual model and the conceptual models of its neighbors in order to be able to translate the received queries to queries that can be understood (and thus answered) by the recipient sources. These mapping could be established manually (as in the case of Semantic Web [8]) but the more appropriate approach for a P2P network, and the more challenging, is the *automatic mapping*. For all these reasons, a simple, conceptually clear, and application-independent conceptual modeling approach seems to be advantageous.

In this paper we consider the case where peers employ *taxonomies*. Note that it is quite easy to create a taxonomy for a source or a mediator. Even ordinary Web users can design this kind of conceptual model. Taxonomies can be constructed either from scratch, or by extracting them from existing taxonomies (e.g. from the taxonomy of Yahoo! or ODP) using special-purpose languages and tools (e.g. see [30]). Furthermore, the design of taxonomies can be done more systematically if done following a faceted approach (e.g. see [27, 26]). In addition, thanks to techniques that have emerged recently [31], taxonomies of compound terms can be also defined in a flexible and systematic manner. However, the more important for P2P systems, advantage of taxonomies is that their simplicity and modeling uniformity allows integrating the contents of several sources without having to tackle complex structural differences. Indeed, as it is shown in [32], inter-taxonomy mappings offer a *uniform* method for bridging *naming, contextual* and *granularity* heterogeneities between the taxonomies of the sources. Given this conceptual modeling approach, a mediator does not have to tackle complex structural differences between the sources, as it happens with relational mediators (e.g. see [22, 21]) and Description Logics-based medi-

ators (e.g. see [17, 11]). Moreover, it allows the integration of *schema* and *data* in a uniform manner. Another advantage of this conceptual modeling approach is that query evaluation in taxonomy-based sources and mediators can be done efficiently (polynomial time).

In this paper we introduce a data-driven method for automatic taxonomy articulation. We call this method *ostensive* because the meaning of each term is explained by ostension, i.e. by pointing to something (here, to a set of objects) to which the term applies. For example, the word "rose" can be defined ostensively by pointing to a rose and saying "that is a rose". Instead, the verbal methods of term definition (e.g. the synonyms or the analytic method) presuppose that the learner already knows some other terms and, thus, they are useless to someone who does not know these terms; e.g. verbal word definitions are useless to a small child who has not learnt any words at all.

Specifically, in this paper we describe an ostensive articulation method that can be used for articulating both single terms and queries, and it can be implemented efficiently by a communication protocol. However, ostensive articulation is possible in a P2P system only if the domain of the peers is not disjoint. If it is disjoint then we cannot derive any articulation. This problem can be tackled by employing *reference collections*. For instance, each peer can have its own taxonomy, but before joining the network it must first index the objects of a small reference object set. Consequently, peers can build automatically the desired articulations by running the articulation protocol on this reference collection.

The rest of this paper is organized as follows: Section 2 introduces a general formal framework for ostensive articulation. Section 3 specializes and describes ostensive articulation for taxonomy-based sources. Section 4 discusses the application of ostensive articulation in P2P systems of taxonomy-based sources, and finally, Section 5 concludes the paper.

2 Ostensive Articulation

Let us first introduce the general framework. We view a source S as a function $S : Q \rightarrow \mathcal{A}$ where Q is the set of all queries that S can answer, and $\mathcal{A}$ is the set of all answers, i.e. $\mathcal{A}$=$\{\ S(q) \mid q \in Q\}$. As we focus on retrieval queries, we assume that $\mathcal{A}$ is a subset of $\mathcal{P}(Obj)$ where Obj is the set of all objects stored at the source.

The ostensive articulation technique that we shall introduce requires a "naming service", i.e. a method for computing one (or may more) name (e.g. query) for each set of objects $R \subseteq Obj$. Let Q_N denote the set of all names. In general, $Q_N = Q$, however we introduce Q_N because we may want names to be queries of a specific form. For supporting the naming service we would like a function $n : \mathcal{P}(Obj) \rightarrow Q_N$ such that for each $R \subseteq Obj$, $S(n(R)) = R$. Having such a function, we would say that $n(R)$ is an exact name for R. Note that if S is an *onto* function and $Q_N = Q$, then the naming function n coincides with the inverse relation of S, i.e. with the relation $S^{-1} : \mathcal{P}(Obj) \rightarrow Q$. However, this

is not always the case, as more often than not, S is not an onto function, i.e. $\mathcal{A} \subset \mathcal{P}(Obj)$. For this reason we shall introduce two naming functions, a *lower* naming function n^- and an *upper* naming function n^+. To define these functions, we first need to define an ordering over queries. Given two queries, q and q' in Q, we write $q \leq q'$ if $S(q) \subseteq S(q')$, and we write $q \sim q'$, if both $q \leq q'$ and $q' \leq q$ hold. Note that $\sim$ is an equivalence relation over Q, and let $Q_\sim$ denote the set of equivalence classes induced by $\sim$ over Q. Note that $\leq$ is a partial order over $Q_\sim$.

Now we can define the function n^- and n^+ as follows:

$$n^-(R) = lub\{\ q \in Q_N \mid S(q) \subseteq R\}$$
$$n^+(R) = glb\{\ q \in Q_N \mid S(q) \supseteq R\}$$

where R is any subset of Obj. Now let R be a subset of Obj for which both $n^-(R)$ and $n^+(R)$ are defined (i.e. the above lub and glb exist). It is clear that in this case it holds:

$$S(n^-(R)) \subseteq R \subseteq S(n^+(R))$$

and that $n^-(R)$ and $n^+(R)$ are the best "approximations" of the exact name of R. Note that if $S(n^-(R)) = S(n^+(R)$ then both $n^-(R)$ and $n^+(R)$ are exact names of R.

If Q_N is a query language that (a) supports disjunction ($\vee$) and conjunction ($\wedge$) and is closed with respect to these, and (b) has a top ($\top$) and a bottom ($\bot$) element such that $S(\top) = Obj$ and $S(\bot) = \emptyset$, then the functions n^- and n^+ are defined for every subset R of Obj. Specifically, in this case $(Q_\sim, \leq)$ is a complete lattice, thus these functions are defined as:

$$n^-(R) = \bigvee\{\ q \in Q_N \mid S(q) \subseteq R\}$$
$$n^+(R) = \bigwedge\{\ q \in Q_N \mid S(q) \supseteq R\}$$

As Q_N is usually an infinite language, $n^-(R)$ and $n^+(R)$ are queries of infinite length. This means that in practice we also need for a method for computing a query of finite lenght that is equivalent to $n^-(R)$ and another one that is equivalent to $n^+(R)$.

If however Q_N does not satisfy the above ((a) and (b)) conditions, then $n^-(R)$ and $n^+(R)$ may not exist. For example, this happens if we want to establish relationships between single terms of two taxonomy-based sources, or between atomic concepts of two Description Logics-based sources. For such cases, we can define n^- and n^+ as follows:

$$n^-(R) = max\{\ q \in Q_N \mid S(q) \subseteq R\}$$
$$n^+(R) = min\{\ q \in Q_N \mid S(q) \supseteq R\}$$

where max returns the maximal element(s), and min the minimal(s). Clearly, in this case we may have several lower and upper names for a given R.

We can now proceed and describe the ostensive articulation. Consider two sources $S_i : Q_i \rightarrow \mathcal{P}(Obj_i)$, and $S_j : Q_j \rightarrow \mathcal{P}(Obj_j)$. Ostensive articulation is

possible only if their domains are not disjoint, i.e. if $Obj_i \cap Obj_j \neq \emptyset$. Let C denote their common domain, i.e. $C = Obj_i \cap Obj_j$. The method that we shall describe yields relationships that are extensionally valid in C.

Suppose that S_i wants to establish an articulation $a_{i,j}$ to a source S_j. An articulation $a_{i,j}$ can contain relationships of the form:

(i) $q_i \geq q_j$,
(ii) $q_i \leq q_j$

where $q_i \in Q_i$, $q_j \in Q_j$. These relationships have the following meaning:

(i) $q_i \geq q_j$ means that $S_i(q_i) \cap C \supseteq S_j(q_j) \cap C$
(ii) $q_i \leq q_j$ means that $S_i(q_i) \cap C \subseteq S_j(q_j) \cap C$

Before describing ostensive articulation let us make a couple of remarks. The first is that the form (i or ii) of the relationships of an articulation depends on the internal structure and functioning of the source that uses the articulation. For instance, suppose that S_i acts as a mediator over S_j. If S_i wants to compute *complete* (with respect to C) answers, then it should use only relationships of type (i) during query translation. On the other hand, if S_i wants to compute *sound* (with respect to C) answers then it should use relationships of type (ii) (e.g. see [21]).

Another interesting remark is that if S_i is a mediator that adopts a *global-as-view* modeling approach, then all q_i that appear in $a_{i,j}$ are primitive concepts. On the other hand, if S_i adopts a *local-as-view* approach then all q_j that appear in $a_{i,j}$ are primitive concepts of S_j.

Below we describe ostensive articulation for the more general case where S_i is interested in relationships of both, (i) and (ii), types, and where q_i, q_j can be arbitrary queries. Let n_j^- and n_j^+ be the naming functions of S_j as defined earlier. Also let $S_i^c(q) = S_i(q) \cap C$ and $S_j^c(q) = S_j(q) \cap C$. Now suppose that S_i wants to articulate a query $q_i \in Q_i$. The query q_i should be articulated as follows:

- $q_i \geq n_j^-(S_i^c(q_i))$ if $S_i^c(q_i) \supseteq S_j^c(n_j^-(S_i^c(q_i)))$
- $q_i \leq n_j^-(S_i^c(q_i))$ if $S_i^c(q_i) \subseteq S_j^c(n_j^-(S_i^c(q_i)))$
- $q_i \geq n_j^+(S_i^c(q_i))$ if $S_i^c(q_i) \supseteq S_j^c(n_j^+(S_i^c(q_i)))$
- $q_i \leq n_j^+(S_i^c(q_i))$ if $S_i^c(q_i) \subseteq S_j^c(n_j^+(S_i^c(q_i)))$

Observe the role of the naming functions. S_j instead of checking all queries in Q_j, it just uses its naming functions in order to compute the lower and the upper name of the set $S_i(q_i) \cap C$. Recall that the naming functions (by definition) return the most precise (semantically close) mapping for q_i, thus this is all that we need.

Furthermore, as we shall see below, the above relationships can be obtained without extensive communication. In fact, they can be obtained by a quite simple and efficient (in terms of exchanged messages) distributed protocol. The protocol

S_i : (1) $A := S_i(q_i)$;
(2) $\text{SEND}_{S_i \to S_j}(\text{A})$

S_j : (3) $F := A \setminus Obj_j$
(4) $A := A \cap Obj_j$;
(5) $down := n_j^-(A)$; $Bdown := S_j(down)$;
(6) $up := n_j^+(A)$; $Bup := S_j(up)$;
(7) $\text{SEND}_{S_j \to S_i}(F,\ down, Bdown, up, Bup)$

S_i : (8) If $(A \setminus F) \supseteq (Bdown \cap Obj_i)$ then set $q_i \geq down$;
(9) If $(A \setminus F) \subseteq (Bdown \cap Obj_i)$ then set $q_i \leq down$;
(10) If $(A \setminus F) \supseteq (Bup \cap Obj_i)$ then set $q_i \geq up$;
(11) If $(A \setminus F) \subseteq (Bup \cap Obj_i)$ then set $q_i \leq up$

Fig. 1. The ostensive articulation protocol

is sketched in Figure 1. Note that only two messages have to be exchanged between S_i and S_j for articulating the query q_i.

Another interesting point is that S_i and S_j do not have to a-priori know (or compute) their common domain C, as C is "discovered" during the run of the protocol (this is the reason why S_j stores in F and sends to S_i those terms that do not belong to Obj_j).

In the case where $Q_N \subset Q$, the only difference is that the message that S_j sends to S_i may contain more than one *up* and *down* queries.

A source can run the above protocol in order to articulate one, several or all of its terms (or queries).

3 Ostensive Articulation for Taxonomy-based Sources

Here we shall specialize ostensive articulation for the case of taxonomy-based sources. Examples of this kind of sources include Web Catalogs (like Yahoo!, Open Directory) and Classification Schemes used in Library and Information Science

We view a taxonomy-based source S as a quadruple $S = \langle T, \preceq, I, Q \rangle$ where:

- T is a finite set of names called *terms*, e.g. `Caranies, Birds`.
- $\preceq$ is a reflexive and transitive binary relation over T called *subsumption*, e.g. `Canaries` $\preceq$ `Birds`.
- I is a function $I : T \to \mathcal{P}(Obj)$ called *interpretation* where Obj is a finite set of objects. For example $Obj = \{1, ..., 100\}$ and $I(\texttt{Canaries}) = \{1, 3, 4\}$.
- Q is the set of all queries defined by the grammar $q ::= t \mid q \wedge q' \mid q \vee q' \mid \neg q \mid (q)$ where t is a term in T.

Figure 2 shows an example of a source consisting of 8 terms and 3 objects[2].

We assume that every terminology T also contains two special terms, the *top term*, denoted by $\top$, and the *bottom term*, denoted by $\bot$. The top term subsumes

[2] We illustrate only the Hasse diagram of the subsumption relation.

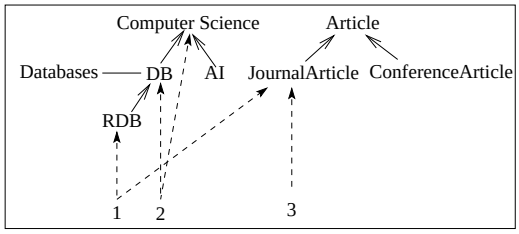

Fig. 2. Graphical representation of a source

every other term t, i.e. $t \preceq \top$. The bottom term is strictly subsumed by every other term t different than top and bottom, i.e. $\bot \preceq \bot$, $\bot \preceq \top$, and $\bot \prec t$, for every t such that $t \neq \top$ and $t \neq \bot$. We also assume that $I(\bot) = \emptyset$ in every interpretation I.

The answer $S(q)$ of a query q is defined as follows (for more see [33]):

$$\begin{aligned} S(t) &= \bigcup\{\ I(t') \mid t' \preceq t\} \\ S(q \wedge q') &= S(q) \cap S(q') \\ S(q \vee q') &= S(q) \cup S(q') \\ S(\neg q) &= Obj \setminus S(q) \end{aligned}$$

For example, in Figure 2 we have $S(\texttt{DB}) = \{1, 2\}$, as $S(\texttt{DB}) = I(\texttt{DB}) \cup I(\texttt{Databases}) \cup I(\texttt{RDB}) = \{1, 2\}$, and $S(\texttt{DB} \wedge \texttt{JournalArticle}) = \{1\}$. We define the *index* of an object o with respect to an interpretation I, denoted by $D_I(o)$, as follows: $D_I(o) = \bigwedge\{t \in T \mid o \in I(t)\}$. For example, in the source of Figure 2 we have $D_I(3) = \texttt{JournalArticle}$ and $D_I(1) = \texttt{RDB} \wedge \texttt{JournalArticle}$.

Let us now define the naming functions for this kind of sources. We define the set of names Q_N as follows: $Q_N = \{\ q \in Q \mid q$ does not contain negation "$\neg$" $\}$. We exclude queries with negation because, as showed in [32], if such queries appear in articulations then we may get systems which do not have a unique minimal model and this makes query evaluation more complicated and less efficient.

The lower and upper name of a set $R \subseteq Obj$ are defined as in the general framework and clearly $(Q_N, \leq)$ is a complete lattice. What remains is to find finite length queries that are equivalent to $n^-(R)$ and $n^+(R)$.

Theorem 1.

$$\begin{aligned} n^-(R) &\sim \bigvee\{\ D_I(o) \mid o \in R,\ S(D_I(o)) \subseteq R\} \\ n^+(R) &\sim \bigvee\{\ D_I(o) \mid o \in R\} \end{aligned}$$

The proof is given in [34]. It is clear that the above queries have finite length, hence they are the queries that we are looking for. For this purpose, hereafter we shall use $n^-(R)$ and $n^+(R)$ to denote the above queries. Note that if the set $\{o \in R,\ S(D_I(o)) \subseteq R\}$ is empty then we consider that $n(R)^- = \bot$. Some examples from the source shown in Figure 3 follow:

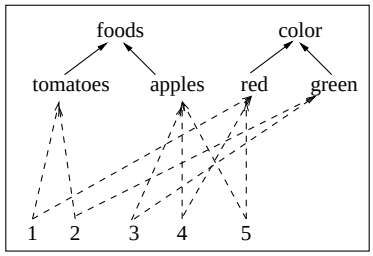

Fig. 3. Example of a source

$$n^+(\{1,3\}) = (\texttt{tomatoes} \wedge \texttt{red}) \vee (\texttt{apples} \wedge \texttt{green})$$
$$n^-(\{1,3\}) = (\texttt{tomatoes} \wedge \texttt{red}) \vee (\texttt{apples} \wedge \texttt{green})$$
$$n^+(\{1,3,5\}) = (\texttt{tomatoes} \wedge \texttt{red}) \vee (\texttt{apples} \wedge \texttt{green}) \vee (\texttt{apples} \wedge \texttt{red})$$
$$n^-(\{1,3,5\}) = (\texttt{tomatoes} \wedge \texttt{red}) \vee (\texttt{apples} \wedge \texttt{green})$$

Let us now demonstrate the articulation protocol for taxonomy-based sources. Consider the sources shown in Figure 4 and suppose that S_1 wants to articulate its terms with queries of S_2. In the following examples we omit the set F (from the message of line (7) of Figure 1) as it is always empty.

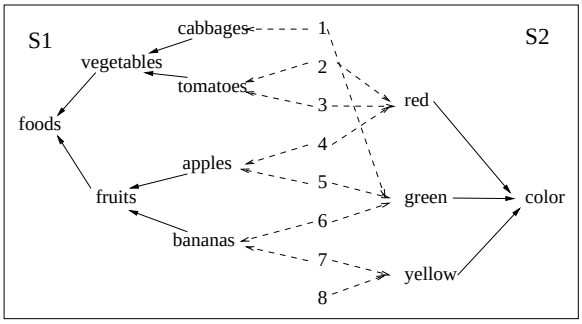

Fig. 4. An example of two sources S_1 and S_2

The steps for articulating the term `cabbages` follow:

$S_1 \rightarrow S_2$: $\{1\}$
$S_2 \rightarrow S_1$: $(\perp, \emptyset)$,(`green`, {1,5,6})
S_1 : `cabbages` $\preceq$ `green`

The steps for articulating the term `apples` follow:

$S_1 \rightarrow S_2$: $\{4, 5\}$
$S_2 \rightarrow S_1$: $(\perp, \emptyset)$, (`red` $\vee$ `green`,{1,2,3,4,5,6})
S_1 : `apples` $\preceq$ `red` $\vee$ `green`

The steps for articulating the term `foods` follow:

$S_1 \rightarrow S_2$: {1,2,3,4,5,6,7}
$S_2 \rightarrow S_1$: (red $\vee$ green,{1,2,3,4,5,6}),
(red $\vee$ green $\vee$ yellow,{1,2,3,4,5,6,7,8})
S_1 : foods $\succeq$ red $\vee$ green,
foods $\sim$ red $\vee$ green $\vee$ yellow

If S_1 runs the protocol for each term of its taxonomy, it will infer the following relationships:

$$\begin{aligned}
\texttt{cabbages} &\preceq \texttt{green}\\
\texttt{tomatoes} &\preceq \texttt{red}\\
\texttt{apples} &\preceq \texttt{red} \vee \texttt{green}\\
\texttt{bananas} &\preceq \texttt{green} \vee \texttt{yellow}\\
\texttt{vegetables} &\preceq \texttt{green} \vee \texttt{red}\\
\texttt{fruits} &\preceq \texttt{red} \vee \texttt{green} \vee \texttt{yellow}\\
\texttt{foods} &\succeq \texttt{red} \vee \texttt{green}\\
\texttt{foods} &\sim \texttt{red} \vee \texttt{green} \vee \texttt{yellow}
\end{aligned}$$

If S_2 runs this protocol for each term of its taxonomy, it will infer the following relationships:

$$\begin{aligned}
\texttt{red} &\succeq \texttt{tomatoes}\\
\texttt{red} &\preceq \texttt{tomatoes} \vee \texttt{apples}\\
\texttt{green} &\succeq \texttt{cabbages}\\
\texttt{green} &\preceq \texttt{cabbages} \vee \texttt{apples} \vee \texttt{bananas}\\
\texttt{yellow} &\preceq \texttt{bananas}\\
\texttt{color} &\sim \texttt{cabbages} \vee \texttt{tomatoes} \vee \texttt{apples} \vee \texttt{bananas}
\end{aligned}$$

The protocol can be used not only for articulating single terms to queries, but also for articulating queries to queries. For example, the steps for articulating the query apples $\vee$ bananas follow:

$S_1 \rightarrow S_2$: {4, 5, 6, 7}
$S_2 \rightarrow S_1$: (red $\vee$ green $\vee$ yellow, {1,2,3,4,5,6,7,8})
S_1 : apples $\vee$ bananas $\preceq$ red $\vee$ green $\vee$ yellow

Now consider the case where we do not want to articulate terms with queries, but terms with *single terms* only, i.e. consider the case where $Q_N = T$. Note that now $lub\{t \in T \mid S(t) \subseteq R\}$ and $glb\{t \in T \mid S(t) \supseteq R\}$ do not always exist. For example, consider the source shown in Figure 5.(a). Note that $n^+(\{1\}) = glb\{t, t'\}$ which does not exist. For the source shown in Figure 5.(b) note that $n^-(\{1,2\}) = lub\{t, t'\}$ which does not exist. Therefore, we can define the upper and lower *names* of a set R as follows: $n^-(R) = max(\{t \in T \mid S(t) \subseteq R)\})$ and $n^+(R) = min(\{t \in T \mid S(t) \supseteq R)\})$. Consider for example the source shown in Figure 5.(c). Here we have:

$$\begin{aligned}
n^-(\{1,2,3\}) &= max(\{c,d,e,b\}) = \{b\}\\
n^+(\{1,2,3\}) &= min(\{b,a\}) = \{b\}
\end{aligned}$$

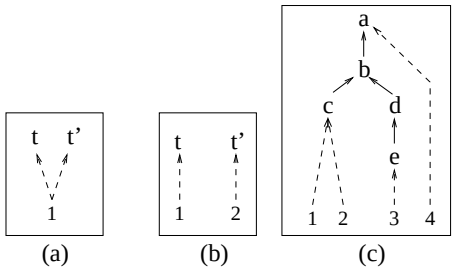

Fig. 5. An example of three sources

Certainly, the relationships obtained by the term-to-term articulation are less expressive than the relationships obtained by the term-to-queries articulation. For instance, suppose that we want to articulate the terms of the source S_1 in each one of the three examples that are shown in Figure 6. Table 1 shows the articulation $a_{1,2}$ that is derived by the *term-to-term* articulation and the *term-to-queries* articulation in each of these three examples.

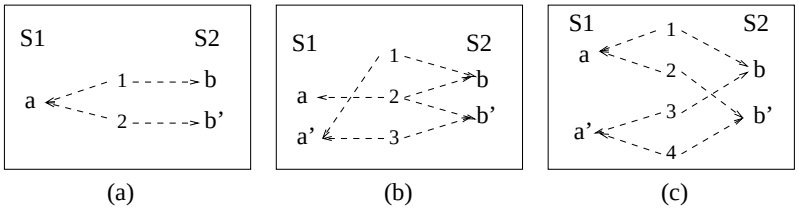

Fig. 6. Three examples

Example	$a_{1,2}$	
	term-to-term art.	*term-to-query* art.
Figure 6.(a)	$a \succeq b$ $a \succeq b'$	$a \sim b \vee b'$
Figure 6.(b)	$a \preceq b$ $a \preceq b'$	$a \sim b \wedge b'$ $a' \preceq b \vee b'$
Figure 6.(c)		$a \preceq b \vee b'$ $a' \preceq b \vee b'$

Table 1. *Term-to-term* vs *term-to-query* articulation

4 Ostensive Articulation in Taxonomy-based P2P Systems

We demonstrated how ostensive articulation can be applied on taxonomy-based sources for constructing inter-taxonomy articulations. Ostensive articulation is

possible in a P2P system only if the domain of the peers is not disjoint. We also assume that every object of Obj has the same identity (e.g. object identifier, URI) in all sources. For domains where no accepted identity/naming standards exist, mapping tables such as the ones proposed in [18] can be employed to tackle this problem. Also techniques from the area of information fusion (that aim at recognizing different objects that represent the same reality) could be also employed for the same purpose. If however the domain of the peers is disjoint then we cannot derive any articulation. One method to tackle this problem is to employ *reference collections*. For instance, each peer can have its own taxonomy, but before joining the network it must first index the objects of a small object set. Consequently, peers can build automatically the desired articulations by running the articulation protocol on this reference collection. Running the protocol on the reference collection C means that the sources S_1 and S_2 instead of using $S_1(q_1)$ and $S_2(q_2)$, they use $S_1(q_1) \cap C$ and $S_2(q_2) \cap C$ respectively. Also note that the employment of reference collections can: (a) *enhance the accuracy* of the resulting articulation, and/or (b) *enhance efficiency*. For instance, if C corresponds to a well known, thus well-indexed set of objects then it can improve the quality of the obtained articulations. For example in the case where S_1 and S_2 are bibliographic sources, C can be a set of 100 famous papers in computer science. A reference collection can also enhance the efficiency of the protocol since a smaller number of objects go back and forth. This is very important, especially in P2P where involved sources are distant.

In a P2P system of taxonomy-based sources, a source apart from object queries now accepts content-based queries, i.e. queries (e.g. boolean expressions) expressed in terms of its taxonomy. For answering a query a source may have to query the neighbor sources. The role of articulations during query evaluation has been described in [33] (for the mediator paradigm) and in [32] (for the P2P paradigm). Roughly, a source in a P2P network can serve any or all of the following roles: primary source, mediator, and query initiator. As a *primary* source it provides original content to the system and is the authoritative source of that data. Specifically, it consists of a taxonomy (i.e. a pair $(T, \preceq)$) plus an object base (i.e. an interpretation I) that describes a set of objects (Obj) in terms of the taxonomy. As a *mediator* it has a taxonomy but does not store or provide any content: its role is to provide a uniform query interface to other sources, i.e. it forwards the received queries after first selecting the sources to be queried and formulating the query to be sent to each one of them. These tasks are determined by the articulations of the mediator. As a *query initiator* it acts as client in the system and poses new queries. Figure 7 sketches graphically the architecture of a network consisting of four peers $S_1, ..., S_4$; two primary sources (S_3 and S_4), one mediator (S_2) and one source that is both primary and mediator (S_1). Triangles denote taxonomies, cylinders object bases, and circles inter-taxonomy mappings. S_2 is a mediator over S_1, S_3 and S_4, while S_1 is a mediator over S_2 and S_3. For more about this architecture and the associated semantics and query evaluation methods please refer to [32].

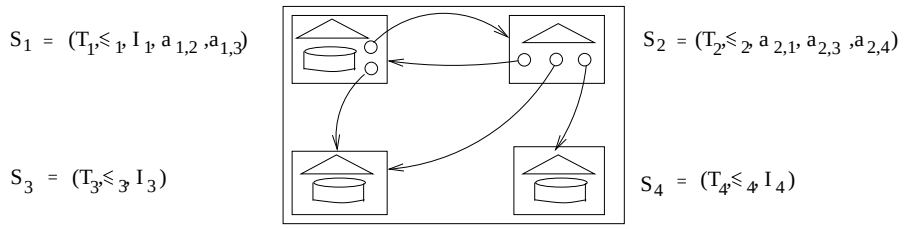

Fig. 7. A P2P network based on taxonomies and inter-taxonomy mappings

5 Conclusion

The contribution of this paper is a formal framework for ostensive data-driven articulation. Roughly, the approaches for linking two conceptual models or taxonomies can be broadly classified as either *model*-driven or *data*-driven.

The model-driven approach starts with a (theoretical) model of how the two taxonomies are constructed and how they are used. Subsequently, the mapping approaches have to address the compatibility, structural and semantic differences and heterogeneities that exist. This is done using software tools (that usually rely on lexical resources) that assist the designer during the articulation process (e.g. see [25, 29, 5, 24]).

On the other hand, in the *data-driven* approach the mappings are *discovered* by examining how terms are used in indexing the objects. The advantage of such an approach is that it does not make any assumptions on how the two taxonomies are constructed, or how they are used. All it requires is the presence of two databases that contain several objects in common. However, the data-driven approach does have inherent difficulties. First, unless one has a large collection of objects that have been indexed using *both* taxonomies, spurious correlation can result in inappropriate linking. Second, if a term is not assigned to any of the common objects, one cannot establish a link for that term. Third, rarely occurring terms can result in statistically insignificant links. Finally, the validation of data-driven approaches can only be statistical in nature. In spite of these inherent difficulties, data-driven approaches can be formalized and automated. However, most of the data-driven approaches that can be found in the literature are applicable only if the domain is a set of documents (texts) (e.g. [6, 16, 12, 20, 28]), and they cannot establish mappings between queries.

The technique described in this paper is quite general and expressive as it can be used for articulating not only single terms but also queries. Furthermore, it can be used for articulating the desired set of terms or queries (it is not obligatory to articulate the entire taxonomies). Another distinctive feature of this technique is that it can be implemented efficiently by a communication protocol, thus the involved sources do not have to reside on the same machine. Therefore it seems appropriate for automatic articulation in P2P systems which is probably the more challenging issue in P2P computing [9].

We also demonstrated how it can be applied to taxonomy-based sources. An interesting remark is that the proposed method can be applied not only to manually constructed taxonomies but also to taxonomies derived automatically on the basis of an inference service. For instance, it can be applied on sources

indexed using taxonomies of compound terms which are defined algebraically [31]. Furthermore it can be applied on concept lattices formed using Description Logics (DL) [13].

One issue for further research, is to investigate how a source that wants to articulate a set $F \subseteq Q$ must use the described protocol in order to obtain the desired articulation with the minimal number of exchanged messages and the less network throughput. Another issue for further research is to investigate ostensive articulation for other kinds of sources.

Acknowledgements

The first author wants to thank his wife Tonia for being an endless source of happiness and inspiration.

References

1. "About LEGION - The Grid OS"(www.appliedmeta.com/legion/about.html), 2000.
2. "How Entropia Works"(www.entropia.com/how.asp), 2000.
3. "Groove" (www.groove.net), 2001.
4. "Napster"(www.naptster.com), 2001.
5. Bernd Amann and Irini Fundulaki. "Integrating Ontologies and Thesauri to Build RDF Schemas". In *Proceedings of the Third European Conference for Digital Libraries ECDL'99*, Paris, France, 1999.
6. S. Amba. "Automatic Linking of Thesauri". In *Proceeding of SIGIR'96*, Zurich, Switzerland, 1996. ACM Press.
7. T.E. Anderson, M. Dahlin, J. M. Neefe, D. A. Patterson, D. S. Roselli, and R. Wang. "Serveless Network File Systems". *SOSP*, 29(5), 1995.
8. Tim Berners-Lee, James Hendler, and Ora Lassila. "The Semantic Web". *Scientific American*, May 2001.
9. Philip A. Bernstein, F. Giunchiglia, A. Kementsietsidis, J. Mylopoulos, L. Serafini, and I. Zaihrayeu. "Data Management for Peer-to-Peer Computing: A Vision". In *Proceedings of WebDB02*, Madison, Wisconsin, June 2002.
10. W. J. Bolosky, J. R. Douceur, D. Ely, and M. Theimer. "Feasibility of a Serveless Distributed File System Deployed on an Existing Set of Desktop PCs". In *Proceedings of Measurement and Modeling of Computer Systems*, June 2000.
11. Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini. A framework for ontology integration. In *Proc. of the 2001 Int. Semantic Web Working Symposium (SWWS 2001)*, pages 303–316, 2001.
12. A. Doan, J. Madhavan, P. Domingos, and A. Halevy. "Learning to Map between Ontologies on the Semantic Web". In *Proceedings of the World-Wide Web Conference (WWW-2002)*, 2002.
13. F.M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. *"Reasoning in Description Logics"*, chapter 1. CSLI Publications, 1997.
14. Steven Gribble, Alon Halevy, Zachary Ives, Maya Rodrig, and Dan Suiu. "What can Databases do for Peer-to-Peer?". In *Proceedings of WebDB01*, Santa Barbara, CA, 2001.

15. Alon Halevy, Zachary Ives, Peter Mork, and Igor Tatarinov. "Piazza: Data Management Infrastructure for Semantic Web Applications". In *Proceedings of WWW'2003*, May 2003.
16. Heiko Helleg, Jurgen Krause, Thomas Mandl, Jutta Marx, Matthias Muller, Peter Mutschke, and Robert Strogen. "Treatment of Semantic Heterogeneity in Information Retrieval". Technical Report 23, Social Science Information Centre, May 2001. (http://www.gesis.org/en/publications/ reports/iz_working_papers/).
17. Vipul Kashyap and Amit Sheth. "Semantic Heterogeneity in Global Information Systems: the Role of Metadata, Context and Ontologies ". In *Cooperative Information Systems: Trends and Directions*. Academic Press, 1998.
18. A. Kementsietsidis, Marcelo Arenas, and Rene J. Miller. "Mapping Data in Peer-to-Peer Systems: Semantics and Algorithmic Issues". In *Int. Conf. on Management of Data, SIGMOD'2003*, San Diego, California, June 2003.
19. J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao. "Oceanstore: An Architecture for Global-Scale Persistent Storage". In *ASPLOS*, November 2000.
20. M. Lacher and G. Groh. "Facilitating the Exchange of Explicit Knowledge Through Ontology Mappings". In *Proceedings of the 14th Int. FLAIRS Conference*, 2001.
21. Maurizio Lenzerini. Data integration: A theoretical perspective. In *Proc. ACM PODS 2002*, pages 233–246, Madison, Wisconsin, USA, June 2002.
22. Alon Y. Levy. "Answering Queries Using Views: A Survey". *VLDB Journal*, 2001.
23. Bo Ling, Zhiguo Lu, Wee Siong Ng, BengChin Ooi, Kian-Lee Tan, and Aoying Zhou. "A Content-Based Resource Location Mechanism in PeerIS". In *Proceedings of the 3rd International Conference on Web Information Systems Engineering, WISE 2002*, Singapore, December 2002.
24. Bernardo Magnini, Luciano Serafini, and Manuela Speranza. "Making Explicit the Hidden Semantics of Hierarchical Classification". In *Atti dell'Ottavo Congresso Nazionale dell'Associazione Italiana per l'Intelligenza Artificiale, LNCS. Springer Verlag*, 2003.
25. P. Mitra, G. Wiederhold, and J. Jannink. "Semi-automatic Integration of Knowledge sources". In *Proc. of the 2nd Int. Conf. On Information FUSION*, 1999.
26. Ruben Prieto-Diaz. "Implementing Faceted Classification for Software Reuse". *Communications of the ACM*, 34(5), 1991.
27. S. R. Ranganathan. "The Colon Classification". In Susan Artandi, editor, *Vol IV of the Rutgers Series on Systems for the Intellectual Organization of Information*. New Brunswick, NJ: Graduate School of Library Science, Rutgers University, 1965.
28. I. Ryutaro, T. Hideaki, and H. Shinichi. "Rule Induction for Concept Hierarchy Allignment". In *Proceedings of the 2nd Workshop on Ontology Learning at the 17th Int. Conf. on AI (IJCAI)*, 2001.
29. Marios Sintichakis and Panos Constantopoulos. "A Method for Monolingual Thesauri Merging". In *Proceedings of 20th International Conference on Research and Development in Information Retrieval, ACM SIGIR'97*, Philadelphia, PA, USA, July 1997.
30. Nicolas Spyratos, Yannis Tzitzikas, and Vassilis Christophides. "On Personalizing the Catalogs of Web Portals". In *15th International FLAIRS Conference, FLAIRS'02*, Pensacola, Florida, May 2002.
31. Yannis Tzitzikas, Anastasia Analyti, Nicolas Spyratos, and Panos Constantopoulos. "An Algebraic Approach for Specifying Compound Terms in Faceted Taxonomies". In *13th European-Japanese Conference on Information Modelling and Knowledge Bases*, Kitakyushu, Japan, June 2003.

32. Yannis Tzitzikas, Carlo Meghini, and Nicolas Spyratos. "Taxonomy-based Conceptual Modeling for Peer-to-Peer Networks". In *Proceedings of 22th Int. Conf. on Conceptual Modeling, ER'2003*, Chicago, Illinois, October 2003.
33. Yannis Tzitzikas, Nicolas Spyratos, and Panos Constantopoulos. "Mediators over Ontology-based Information Sources". In *Second International Conference on Web Information Systems Engineering, WISE 2001*, Kyoto, Japan, December 2001.
34. Yannis T. Tzitzikas. *"Collaborative Ontology-based Information Indexing and Retrieval"*. PhD thesis, Department of Computer Science - University of Crete, September 2002.