Procs. of ER'2003, Chicago, Illinois, Oct. 13-16, 2003. LNCS, © *Springer-Verlag*

Taxonomy-based Conceptual Modeling for Peer-to-Peer Networks

Yannis Tzitzikas[1,2] Carlo Meghini[1] Nicolas Spyratos[3]

[1] *Istituto di Scienza e Tecnologie dell' Informazione [ISTI]*
Consiglio Nazionale delle Ricerche [CNR], Pisa, Italy
Email : {tzitzik,meghini}@isti.cnr.it
[3] *Laboratoire de Recherche en Informatique, Universite de Paris-Sud, France*
Email : spyratos@lri.fr

Abstract. We present a taxonomy-based conceptual modeling approach for building P2P systems that support semantic-based retrieval services. We adopt this simple conceptual modeling approach due to its advantages in terms of ease of use, uniformity, scalability and efficiency. As each peer uses its own taxonomy for describing the contents of its objects and for formulating queries to the other peers, peers are equipped with *articulations*, i.e. inter-taxonomy mappings, in order to carry out the required translation tasks. We describe various kinds of articulations and we give the semantics for each case. Then we discuss the differences between query evaluation in mediators and query evaluation in P2P systems, and finally we identify issues for further research.

1 Introduction

There is a growing research and industrial interest on peer-to-peer (P2P) systems. A peer-to-peer 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 (e.g. Napster), storage services [7, 20], or distributed "grid" computation (e.g. Entropia, Legion). Smaller-scale P2P systems also exist, such as federated, server-less file systems [3, 2] and collaborative workgroup tools (e.g. Groove). 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 networks is a great challenge that raises questions about data models, conceptual modeling, query languages and techniques for query evaluation and dynamic schema mapping. Roughly, 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 former 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. [10]). In this paper we focus on the latter case where the objects of a peer are indexed according to a specific conceptual model represented in a particular data model (e.g. relational, object-oriented, logic-based, etc), and content searches are formulated using a specific query language. 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 (e.g. see [13]), so that traditional information mediation and integration techniques will likely apply (with the

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

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.

In this paper we propose an approach which is based on *taxonomies*. Taxonomies are very easy to build in comparison to other kinds of conceptual models. Moreover, as we shall see, if the conceptualization of the domain is a set of objects (e.g. a set of music files, images, etc) then this approach is as expressive as other more sophisticated conceptual modeling approaches. Peers can construct their taxonomies either from scratch or by extracting them from existing taxonomies (e.g. from the taxonomy of Open Directory or Yahoo!) using special-purpose languages and tools (e.g. like the one presented in [19]). In addition to its taxonomy, a source can have an object base, i.e. a database that indexes the objects of the domain under the terms of the taxonomy. Information integration, reconcilement and personalization is achieved through *mediators*, i.e. through sources which in addition to their taxonomies are enriched with *articulations* to the other sources of the network, where an articulation is actually a mapping between the terms of the mediator and the terms of the sources. Of course, in a pure P2P system we cannot partition sources to primary and secondary (i.e. mediators) as we may have mutually articulated sources. We describe the semantics of P2P systems of this kind and we identify issues that require further research concerning query evaluation and optimization.

The remaining of this paper is organized as follows: Section 2 discusses the benefits of taxonomies with respect to the P2P paradigm. Section 3 describes the building blocks of a network of articulated sources, i.e. sources, mediators and articulated sources. Section 4 describes the semantics of the network. Section 5 discusses query evaluation issues. Finally, Section 6 concludes the paper and identifies issues for further research.

2 Taxonomies

Taxonomies is probably the oldest conceptual modeling tool. Nevertheless, it is a powerful tool still used is in libraries, in very large collections of objects (e.g. see [17]) and the Web (e.g. Yahoo!, Open Directory). Although more sophisticated conceptual models (including concepts, attributes, relations and axioms) have emerged and are recently employed even for meta-tagging in the Web [11, 25], almost all of them have a backbone consisting of a subsumption hierarchy, i.e. a taxonomy.

What we want to emphasize here, is that in a very broad domain, such as the set of all Web pages or in large scale P2P system, it is not easy to identify the classes of the domain because the domain is too wide and different users, or applications, conceptualize it differently, e.g. one class of the conceptual model according to one user may correspond to a value of an attribute of a class of the conceptual model according to another user. For example, Figure 1 shows two different conceptual models for the same domain. We consider only two objects of the domain, denoted by the natural numbers 1 and 2. The conceptual model of Figure 1.(a) is appropriate for building an information system for a furniture store, while the conceptual model of Figure 1.(b) is appropriate for building an information system for a department store. The classes of model (a), i.e. the classes `Tables`, `Chairs` and `Couches`, have been defined so as to distinguish the objects of the domain according to their *use*. On the other hand, the classes of model (b), i.e. the classes `Wooden`, `Plastic` and `Glassware`, have been defined so as to distinguish the objects of the domain according to their *material*. This kind of distinction is useful for a department store, as it determines (up to some degree) the placement of the objects in the various departments of the store.

Figure 2 shows a taxonomy for the same domain which consists of terms and subsumption links only. This taxonomy seems to be more application independent. All criteria (characteristics) for distinguishing the objects are equally "honoured".

A simple conceptual modeling approach where each conceptual model is a taxonomy, has two main advantages. The first is that it is very easy to create the conceptual model of a source or a mediator. Even ordinary Web users can design this kind of conceptual models. Furthermore, the design can be done more systematically if done following a faceted approach (e.g. see [15, 14]). In addition, thanks to techniques that have emerged recently [21], taxonomies of compound terms can be also defined in a flexible and systematic manner.

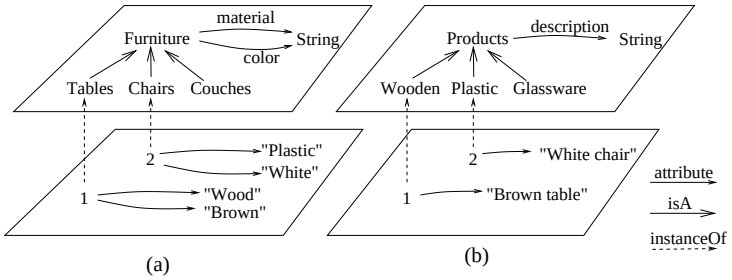


Fig. 1. Two different conceptual models for the same domain

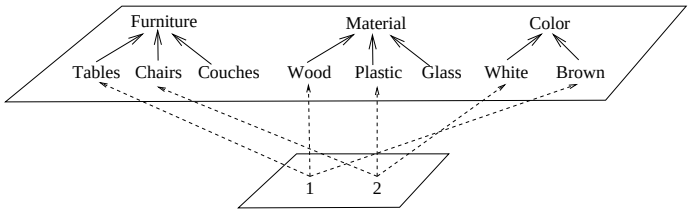


Fig. 2. A taxonomy that consists of terms and subsumption links only

The second, and more important for P2P systems, advantage is that the simplicity and modeling uniformity of taxonomies allows integrating the contents of several sources without having to tackle complex structural differences. Indeed, as it will be seen in the subsequent sections inter-taxonomy mappings offer a *uniform* method to bridge *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 happens with relational mediators (e.g. see [9, 8]) and Description Logics-based mediators (e.g. see [6, 4]). 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).

Due to the above benefits (conceptual modeling simplicity, integration flexibility, query evaluation efficiently), taxonomies seem appropriate for large scale pure P2P systems. The only assumption that we make is that the domain is a set of objects which we want to index and subsequently retrieve, without being interested in the relationships that may hold between the objects of the domain.

3 The Network

Let *Obj* denote the set of all objects of a domain common to several sources. A typical example of such a domain is the set of all pointers to Web pages. A network of articulated sources over *Obj* is a set of sources $U = \{S_1, ...S_n\}$ where each S_i falls into one of the following categories:

Simple sources: they consist of a taxonomy and an object base, i.e. a database that indexes objects of *Obj* under the terms of the taxonomy. A simple source accepts queries over its taxonomy and returns the objects whose index "matches" the query.

Mediators: they consist of a taxonomy plus a number of articulations to other sources of the network. Again, a mediator accepts queries over its taxonomy but as it does not maintain an object base, query answering requires sending queries to the underlying sources and combining the returned results.

Articulated sources: they are both simple sources and mediators, i.e. they consist of a taxonomy, an object base and a number of articulations to other sources of the network. An articulated source

can behave like a simple source, like a mediator, or like a mediator which in addition to the external sources can also use its own simple source for query answering.

Clearly, simple sources and mediators are special cases (or roles) of articulated sources. Each kind of source is described in detail below.

3.1 Simple Sources

A simple source S is a pair $\langle (T, \preceq), I \rangle$ where $(T, \preceq)$ is a taxonomy and I is an interpretation of T.

Definition 1. A taxonomy is a pair $(T, \preceq)$ where T is a *terminology*, i.e. a finite and non empty set of names, or *terms*, and $\preceq$ is a reflexive and transitive relation over T called *subsumption*.

If a and b are terms of T and $a \preceq b$ we say that a is *subsumed* by b, or that b *subsumes* a; for example, `Databases` $\preceq$ `Informatics`, `Canaries` $\preceq$ `Birds`. We say that two terms a and b are *equivalent*, and write $a \sim b$, if both $a \preceq b$ and $b \preceq a$ hold, e.g., `Computer Science` $\sim$ `Informatics`. Note that the subsumption relation is a preorder over T and that $\sim$ is an equivalence relation over the terms T. Moreover $\preceq$ is a partial order over the equivalence classes of terms induced by $\sim$.

In addition to its taxonomy, each source has a stored *interpretation* I of its terminology, i.e. a total function $I : T \rightarrow 2^{Obj}$ that associates each term of T with a set of objects. Here, we use the symbol 2^{Obj} to denote the powerset of Obj. Figure 3 shows an example of a simple source. In this and subsequent figures the objects are represented by natural numbers and membership of objects to the interpretation of a term is indicated by a dotted arrow from the object to that term. For example, the objects 1 and 3 in Figure 3 are members of the interpretation of the term `JournalArticle`, i.e. $I($`JournalArticle`$) = \{1, 3\}$. Subsumption of terms is indicated by a continuous-line arrow from the subsumed term to the subsuming term. Note that we do not represent the entire subsumption relation but its Hasse diagram, in which the reflexive and the transitive arrows are omitted. Equivalence of terms is indicated by a continuous non-oriented line segment. Note that equivalence captures the notion of synonymy, and that each equivalence class simply contains alternative terms for naming a set of objects.

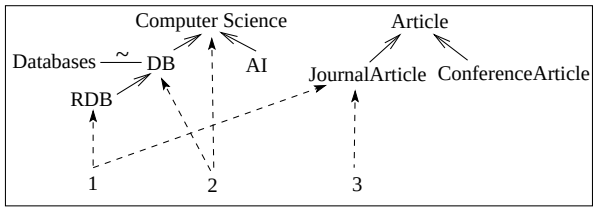


Fig. 3. Graphical representation of a source

3.2 Mediators and Articulated Sources

A *mediator* is a secondary source that bridges the heterogeneities that may exist between two or more sources in order to provide a uniform query interface to an integrated view of these sources. According to the model presented in [23], a mediator has a taxonomy with terminology and structuring that reflects the needs of its potential users, but does *not* maintain a database of objects. Instead, the mediator maintains a number of *articulations* to the sources. An articulation to a source is a set of relationships between the terms of the mediator and the terms of that source.

These relationships can be defined manually by the designer of the mediator, or they can be constructed automatically or semi-automatically following a model-driven approach (e.g. [18, 12]) or a data-driven approach (e.g. [22, 5, 16, 1]). In this paper we do not focus on articulation design or construction (we treat this issue in [22]).

Users formulate queries over the taxonomy of the mediator and it is the task of the mediator to choose the sources to be queried, and to formulate the query to be sent to each source. To this end, the mediator uses the articulations in order to translate queries over its own taxonomy to queries over the taxonomies of the articulated sources. Then it is again the task of the mediator to combine appropriately the results returned by the sources in order to produce the final answer.

Definition 2. An *articulation* from a taxonomy $(T_i, \preceq_i)$ to a taxonomy $(T_j, \preceq_j)$, denoted by $\preceq_{a_{i,j}}$, or just $a_{i,j}$, is any set of relationships $t_j \preceq t_i$ where $t_i \in T_i$ and $t_j \in T_j$.

We assume that each term has a unique identity over the network and that all terminologies are pairwise disjoint.

Definition 3. A *mediator* M over k sources $S_1, ..., S_k$ consists of:
1) a taxonomy $(T_M, \preceq_M)$, and
2) a set $\{a_{M,1}, ..., a_{M,k}\}$, where each $a_{M,i}$ is an *articulation* from $(T_M, \preceq_M)$ to $(T_i, \preceq_i)$.

Figure 4.(a) shows an example of a mediator over two sources that provide access to electronic products. The articulation $a_{M,1}$ shown in this figure is the following sets of subsumption relationships: $a_{M,1} = \{\texttt{PhotoCameras}_1 \preceq \texttt{Cameras}, \texttt{Miniature}_1 \preceq \texttt{StillCameras}, \texttt{Instant}_1 \preceq \texttt{StillCameras}, \texttt{Reflex}_1 \preceq \texttt{Reflex}\}$ while the articulation $a_{M,2}$ is $a_{M,2} = \{\texttt{Products}_2 \preceq \texttt{Electronics}, \texttt{SLRCams}_2 \preceq \texttt{Reflex}, \texttt{VideoCams}_2 \preceq \texttt{MovingPictureCams}\}$.

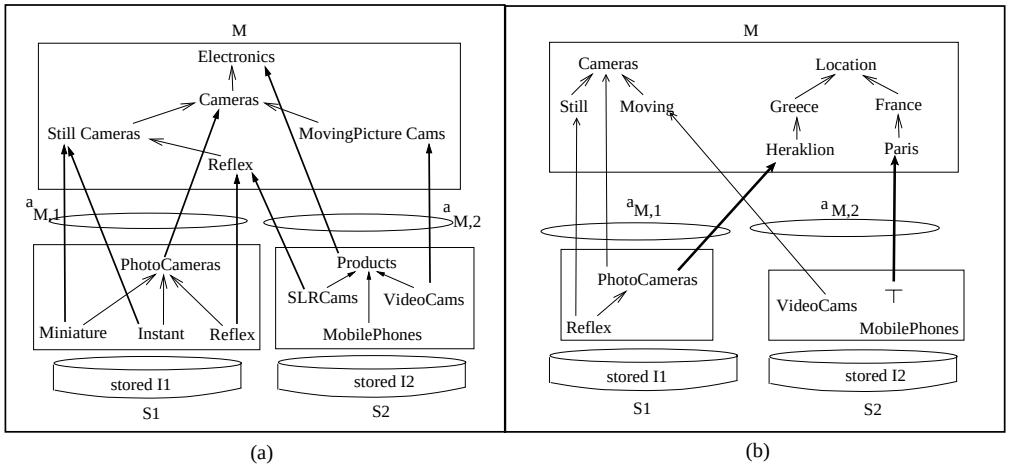


Fig. 4. Two examples of a mediator

Integrating objects from several sources often requires *restoring the context* of these objects, i.e. adding information that is missing from the original representation of the objects which concerns the context of the objects. An example that demonstrates how the articulations can restore the context of the objects is shown in Figure 4.(b). The illustrated mediator provides access to electronic products according to the *type* of the products and according to the *location* of the stores that sell these products. The mediator has two underlying sources S_1 and S_2, where the former corresponds to a store located in Heraklion, while the latter corresponds to a store located in Paris. The context of the objects of each source, here the location of the store that sells each product, can be restored by adding to the articulations appropriate relationships. Specifically, for defining that all `PhotoCameras` of the source S_1 are available through a store located in `Heraklion`, it suffices to put in the articulation $a_{M,1}$ the relationship $\texttt{PhotoCameras}_1 \preceq \texttt{Heraklion}$, while for defining that all products of the source S_2 are available through a store located in `Paris`, it suffices to put

in the articulation $a_{M,2}$ the following relationship $\top_2 \preceq$ `Paris`, where $\top_2$ denotes the maximal element of the subsumption relation of S_2.

An articulated source is a source that is both simple and mediator.

Definition 4. An *articulated source* M over k sources $S_1, ..., S_k$ consists of:
1) a taxonomy $(T_M, \preceq_M)$,
2) a stored interpretation I_s of T_M, and
3) one *articulation* $a_{M,i}$ for each source S_i, $1 \leq i \leq k$.

4 Semantics and Queries

Suppose that a source S receives a query q. In this section we shall give answer to the following question: what answer S should return ? We will answer this question for the cases where S is: (a) a simple source, (b) a mediator, (c) an articulated source, and (d) a source (mediator or articulated source) that participates in a P2P system.

4.1 Simple Sources

A simple source $\langle (T, \preceq), I \rangle$ answers queries based on the stored interpretation of its terminology. However, in order for query answering to make sense, the interpretation that a source uses for answering queries must respect the structure of the source's taxonomy (i.e. the relation $\preceq$) in the following sense: if $t \preceq t'$ then $I(t) \subseteq I(t')$.

Definition 5. An interpretation I is a *model* of a taxonomy $(T, \preceq)$ if for all t, t' in T, if $t \preceq t'$ then $I(t) \subseteq I(t')$.

Definition 6. Given an interpretation I of T we define the model of $(T, \preceq)$ *generated* by I, denoted $\bar{I}$, as follows: $\bar{I}(t) = \bigcup\{I(s) \mid s \preceq t)\}$.

The set of interpretations of a given terminology T can be ordered using pointwise set inclusion, i.e. given two interpretations I, I' of T, we call I less than or equal to I', and we write $I \leq I'$, if $I(t) \subseteq I'(t)$ for each term $t \in T$. Note that $\leq$ is a partial order over interpretations. It can be easily seen that if I is an interpretation of T then $\bar{I}$ is the unique minimal model of $(T, \preceq)$ which is greater than or equal to I.

Definition 7. A *query* over a terminology T is any string derived by the following grammar, where t is a term of T: $q ::= \ t \mid q \wedge q' \mid q \vee q' \mid q \wedge \neg q' \mid (q) \mid \epsilon$ We will denote by Q_T the set of all queries over T.

A simple source responds to queries over its own terminology.

Definition 8. Any interpretation I of T can be extended to an interpretation $\hat{I}$ over the set of all queries in Q_T as follows: $I(q \wedge q') = I(q) \cap I(q'), I(q \vee q') = I(q) \cup I(q'), I(q \wedge \neg q') = I(q) \setminus I(q')$. For brevity we use I to denote both I and its extension over Q_T.

We shall use $ans_i(q)$ to denote the answer that a source S_i will return for the query q, i.e. the set $\bar{I}_i(q)$. Query evaluation in taxonomy-based sources can be done in polynomial time with respect the size of T, specifically the computation of $\bar{I}(q)$ can be done in $O(|T| * |Obj|)$ in the case where the transitive closure of $\leq$ is stored.

4.2 Mediators and Articulated Sources

A mediator $\langle (T_M, \preceq_M), \{a_{M,1}, ..., a_{M,k}\} \rangle$ receives queries (boolean expressions) over its own terminology T_M. As it does not have a stored interpretation of T_M, the mediator answers queries using an interpretation of T_M obtained by *querying* the underlying sources. To proceed we need to introduce some notations. If A is a binary relation over a set T then we shall use A^* to denote the transitive closure of A, e.g. if $A = \{(a, b), (b, c)\}$ then $A^* = \{(a, b), (b, c), (a, c)\}$. If S is a subset of

T then the restriction of A on S, denoted by $A_{|S}$, consists of those relationships in A that relate only elements of S. For example if $T = \{a, b, c\}$, $S = \{a, c\}$ and $A = \{(a, b), (b, c)\}$ then $A_{|S} = \emptyset$ while $A^*_{|S} = \{(a, c)\}$.

We define the *total subsumption* of the mediator, denoted by $\sqsubseteq_M$, by taking the transitive closure of the union of the subsumption relation $\preceq_M$ with all articulations of the mediator, that is: $\sqsubseteq_M = (\ \preceq_M\ \cup a_{M,1} \ldots \cup a_{M,k}\ )^*$. Clearly, $\sqsubseteq_M$ is a subsumption relation over $T_M \cup F$, where F consists of all terms that appear in the articulations and are not elements of T_M ($F \subseteq T_1 \cup ... \cup T_k$). Clearly it holds: $(\sqsubseteq_M)_{|_{T_M}} = \preceq_M$.

We can define an interpretation I of the terminology $T_M \cup F$ as follows:

$$I(t) = \begin{cases} \emptyset & \text{if } t \in T_m \\ ans_i(t) & \text{if } t \in T_i \end{cases}$$

Now let $\bar{I}$ denote the model of the taxonomy $(T_M \cup F, \sqsubseteq_M)$ that is generated by I, and let I_M denote the restriction of $\bar{I}$ on T_M, i.e. $\bar{I}_{|T_M}$. Clearly, I_M is a model of $(T_M, \preceq_M)$, and this is the model that the mediator has to use for answering queries. It can be easily seen that the mediator can compute this as follows:

$$I_M(t) = \bigcup_{i=1}^{k} (\cup\{\ ans_i(s) \mid s \in T_i, s \sqsubseteq_M t\}) \tag{1}$$

where t is a term of T_M. For example the interpretation of term `Cameras` of Figure 4.(a) is computed as follows:
$I_M(\texttt{Cameras}) = (\cup\{\ ans_1(s) \mid s \in T_1, s \sqsubseteq_M \texttt{Cameras}\}) \bigcup (\cup\{\ ans_2(s) \mid s \in T_2, s \sqsubseteq_M \texttt{Cameras}\}) = ans_1(\texttt{PhotoCameras}) \cup ans_1(\texttt{Miniature}) \cup ans_1(\texttt{Instant}) \cup ans_1(\texttt{Reflex}) \cup ans_2(\texttt{VideoCams}) \cup ans_2(\texttt{SLRCams})$

It worths mentioning here that as the articulations contain relationships between single terms these kinds of mappings enjoy the benefits of both *global-as-view* (GAV) and *local-as-view* (LAV) approach (see [4, 8] for a comparison). Specifically, we have (a) the query processing simplicity of the GAV approach, as query processing basically reduces to unfolding the query using the definitions specified in the mapping, so as to translate the query in terms of accesses (i.e. queries) to the sources, and (b) the modeling scalability of the LAV approach, i.e. the addition of a new underlying source does not require changing the previous mappings.

Now, an articulated source behaves like a mediator, except that now, in addition to the k external sources $S_1, ..., S_k$, we have the mediator's own simple source $S_M = \langle (T_M, \preceq_M), I_s \rangle$ acting as a $(k+1)$-th source. The interpretation I_M of T_M that is used for answering queries is defined by

$$I_M(t) = \bigcup_{i=1}^{k+1} (\cup\{\ ans_i(s) \mid s \in T_i, s \sqsubseteq_M t\}) \tag{2}$$

Here we assume that $T_{k+1} = T_M$, $I_{k+1} = I_s$, and thus $ans_{k+1}(t) = \bar{I}_{k+1}(t) = \bigcup\{\ I_s(s) \mid s \preceq_M t\}$.

4.3 Mediators and Articulated Sources in a P2P System

Figure 5 shows an example of a P2P network consisting of four sources $S_1, ..., S_4$; two simple sources (S_3 and S_4), one mediator (S_2) and one articulated source (S_1). In a P2P system we can no longer distinguish sources to primary and secondary (i.e. mediators), as we can have mutually articulated sources, e.g. notice the mutually articulated sources S_1 and S_2. Due to this characteristic, we have to consider the entire network in order to define semantics and query answers.

We can view the entire network as a single simple source. Let T denote the union of the terminologies of all sources in the network, i.e. $T = \bigcup_{i=1}^{n} T_i$. An interpretation of the network is any interpretation of the terminology T, i.e. any function $I : T \rightarrow 2^{Obj}$. At any given time point, we can define *the* interpretation of the network by taking the union of the interpretations that are stored in the sources of the network, i.e. $I = I_1 \cup ... \cup I_n$. Reversely, we can consider that there

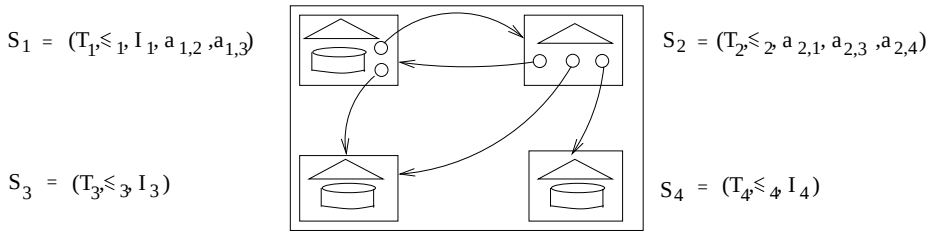


Fig. 5. A network of articulated sources

is one interpretation $I : T \rightarrow 2^{Obj}$ which is stored distributed in k sources $S_1, ..., S_k$ where each source S_i stores a part of I, i.e. a function $I_i : T_i \rightarrow 2^{Obj}$, where the sets $T_1, ..., T_k$ constitute a partition of T. We can define *the* subsumption relation of the network, denoted by $\sqsubseteq$, by taking the union of the total subsumption relations of the sources, i.e.: $\sqsubseteq = (\bigcup_{i=1}^{n} \sqsubseteq_i)^*$.

Now, a model of the network is any model of the taxonomy $(T, \sqsubseteq)$. Analogously to the simple source case, the minimal model that is greater than the interpretation of the network I, denoted by $\bar{I}$, is defined as follows: $\bar{I}(t) = \bigcup\{\ I(t') \mid t' \sqsubseteq t\}$.

It is reasonable to use $\bar{I}$ as the model for answering queries. This means that if a source S_i receives a query q_i over T_i, then it should return the set $\bar{I}(q_i)$. For deriving the set $\bar{I}(t)$, where t is any term of T, each source has to contribute. Specifically, the contribution of each source S_i, denoted by $Contr_i(t)$, is the following: $Contr_i(t) = \bigcup\{\ I_i(t') \mid t' \in T_i,\ t' \sqsubseteq t\}$ Thus we can also write: $\bar{I}(t) = Contr_1(t)\ \cup ... \cup\ Contr_n(t)$.

4.4 Extending the Form of Articulations

Before describing query evaluation in P2P systems let us first study a more general case where an articulation $a_{i,j}$ can contain subsumption relationships between terms of T_i and *queries* in Q_{T_j}. We call such articulations *term-to-query* ($t2q$) articulations and the former *term-to-term* ($t2t$) articulations. Clearly, a $t2q$ articulation can contain relationships that we *cannot* express in a $t2t$ articulation, e.g.: $\texttt{DBArticles}_i \succeq_{a_{i,j}} (\texttt{Databases}_j \wedge \texttt{Articles}_j)$,

$\texttt{FlyingObject}_i \succeq_{a_{i,j}} (\texttt{Birds}_j \wedge \neg\ (\texttt{Ostrich}_j \vee \texttt{Penguin}_j\ ))$.

Formally, a $t2q$ articulation is defined as follows:

Definition 9. A *term-to-query articulation* $a_{i,j}$ is any set of relationships $q_j \preceq t_i$ where $t_i \in T_i$ and $q_j \in Q_{T_j}$.

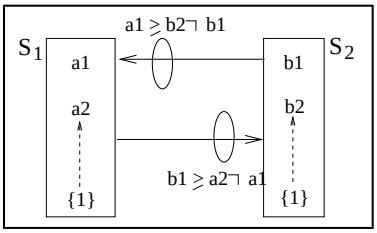
Below we discuss the consequences of this extension. The relation $\sqsubseteq_M$ of a mediator over k sources $S_1, ..., S_k$ is defined similarly to the case of $t2t$ articulations, but now $\sqsubseteq_M$ is a subsumption relation over $T_M \cup Q_{T_1} \cup ... \cup Q_{T_k}$. The interpretation I_M of the mediator can be defined as follows: $I_M(t) = \bigcup_{i=1}^{k}(\cup\{\ ans_i(q) \mid q \in Q_{T_i}, q \sqsubseteq_M\ t\})$. Notice that since the articulations now contain relationships between terms of the mediator and source queries, we are in a *global-as-view* (GAV) approach.

Analogously to the case of $t2t$ articulations, we can view the entire network as one simple source. However, the subsumption relation of the network, i.e. $\sqsubseteq$, now is not a relation over T, but a relation over V, where $V = T \cup \{q \in Q_T \mid q\ \text{ appears in an articulation}\}$. Consequently, the models of the network are defined as:

Definition 10. An interpretation I of T is a *model* of $(V, \sqsubseteq)$ if:
(a) if $t \sqsubseteq t'$ then $I(t) \subseteq I(t')$, and (b) if $q \sqsubseteq t$ then $\hat{I}(q) \subseteq I(t)$.

We can consider that the model of the network is the minimal model of $(V, \sqsubseteq)$ which is greater than the stored interpretation I. If the queries that appear in the articulations do not contain negation, then certainly there is always a unique minimal model. Indeed, we can view the entire

network as a distributed Datalog program whose rules contain only monadic predicates. Specifically, we can view each $o \in I(t)$ as a fact $t(o)$ (where o is a constant), each $t \preceq t'$ as a rule $t'(X)$:- $t(X)$, and each $t2q$ articulation as a set of rules, e.g. the relationship $t \succeq (t_1 \wedge t_2) \vee (t_3 \wedge \neg t_4)$ corresponds to the rules $t(X)$:- $t_1(X), t_2(X)$ and $t(X)$:- $t_3(X), \neg t_4(X)$. It is known that this kind of programs have always a unique minimal model. However, if the queries of the articulations have negation, then the corresponding Datalog program has rules with negation in their bodies, and such programs may not have a unique minimal model (e.g. see [24]). This is also illustrated by the example shown in Figure 6. The table shown in (b) of this figure shows the stored interpretation I of the network and two interpretation which are greater than I, namely I_a and I_b. Note that both are models and both are minimal.



(a)

term/query	I	I_a	I_b
$a1$	$\emptyset$	{**1**}	$\emptyset$
$a2$	{1}	{1}	{1}
$b1$	$\emptyset$	$\emptyset$	{**1**}
$b2$	{1}	{1}	{1}
$b2 \wedge \neg b1$	{1}	{1}	$\emptyset$
$a2 \wedge \neg a1$	{1}	$\emptyset$	{1}

(b)

Fig. 6. A network with $t2q$ articulations which has not a unique minimal model

5 Query Evaluation in P2P Systems

One can easily see that a straightforward query evaluation approach in which each mediator acts without taking into account that it participates in a P2P system is not appropriate as the external cycles (those that contain terms from two or more different sources) in the graph of the relation $\sqsubseteq$ may cause *endless query loops* as it happens in the network shown in Figure 7.(a). Notice the cycle $(a3, c, b4, b2, b1, a3)$ which is also shown in the Hasse diagram of the relation $\sqsubseteq$ that is shown in Figure 7.(a). Let use $q_{i,j}$ to denote a query which is submitted by a source S_i to a source S_j. The first part of the sequence of queries that will be exchanged between the sources, for answering the query $q = a2$, follows: $q_{1,2} = b2,\ q_{2,3} = c,\ q_{3,2} = b3,\ q_{3,1} = a3,\ q_{1,2} = b1,\ q_{2,3} = c$, and so on. Clearly, the query evaluation will never terminate. It can be easily proved that the cycles of $\sqsubseteq$ that contain terms from two or more sources cause this phenomenon.

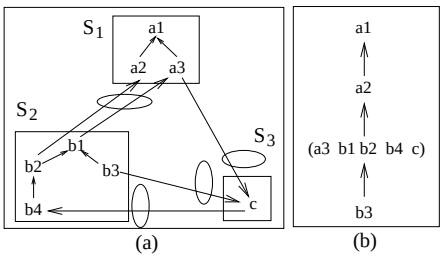


Fig. 7. A network of articulated sources

Let us now consider the networks with $t2q$ articulations with no negation, like the one shown in Figure 8.(a). Notice that endless query loops can arise in this network. Indeed, if S_1 receives

the query $q_1 = a1$ then we will have: $q_{1,2} = b1 \wedge b2$, $q_{2,3} = c1 \wedge c2$, $q_{3,1} = a2$, $q_{1,2} = b1 \wedge b2$, and so on. In networks with $t2t$ articulations we can identify the cases where endless query loops arise using the relation $\sqsubseteq$. However, in our case the relation $\sqsubseteq$ of the network (shown in Figure 8.(b)) does not allow us to detect this phenomenon as $\sqsubseteq$ does not have any external term cycle. For this reason, we can define an auxiliary relation over the set T of the network:

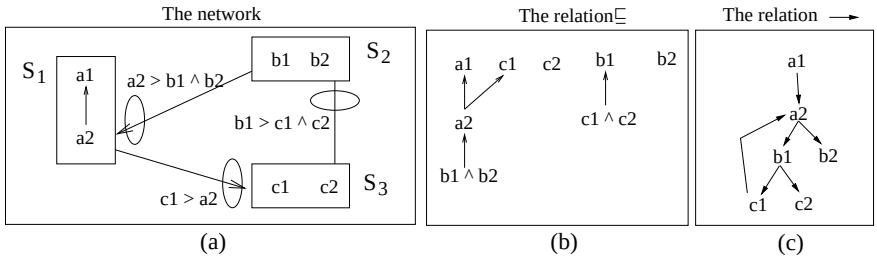


Fig. 8. A network with term-to-query articulations

Definition 11. Let t, t' be two terms of T. We say that t *requires* t' and we write $t \rightarrow t'$, if one of the following holds:
(a) $t \succeq_i t'$ for an $i = 1, .., k$,
(b) $\exists\, q$ such that $t \sqsupseteq q$ and t' appears in q,
(c) $\exists\, t$" such that $t \rightarrow t$" $\rightarrow t'$.

It follows directly from the above definition that if a term t requires a term t' then the computation of $\bar{I}(t)$ requires the computation of $\bar{I}(t')$. This implies that an external term cycle in the relation $\rightarrow$ certainly causes endless query loops. In our example the cycle $a2 \rightarrow b1 \rightarrow c1 \rightarrow a2$ (shown in Figure 8.(c)) is responsible for the endless query loops. Note that in networks with only $t2t$ articulations the relation $\rightarrow$ coincides with the relation $\sqsupseteq$, i.e. $t \rightarrow t'$ iff $t \sqsupseteq t'$.

Below we describe a simple (not optimized) query evaluation method that avoids the endless query loops. At first notice that each source in the network can receive *two* kinds of queries: (a) queries submitted by the *users* of the source, and (b) queries submitted by other *sources* of the network. We may call the former *external queries* and the latter *internal queries*. At first we assume that all external queries are single-term queries. Whenever a source S receives an external query q, it assigns to it a *network-unique identifier* denoted by q_{id}. This identifier will accompany all internal queries that will be exchanged in the network during the evaluation of q. Now consider the following operation mode:

- Each source keeps a log file of the queries that it has received. The log file stores pairs of the form: $(Query, QueryId)$.
- whenever a source S receives an internal query (q', q_{id}) from a source S' which matches a row of the log file, i.e. if (q', q_{id}) is already stored in the log file, then it replies by sending to S' the empty set and it does not query any other underlying source

It can be easily proved that if each source operates in this way, no endless query loops appear and that every external query is answered correctly. This is true for both $t2t$ and $t2q$ articulations with no negation.

However, techniques for reducing the number of queries that have to be exchanged between the sources of the network have to be designed. Furthermore, techniques that allow a source to identify the global relationships, i.e. the relationships of $\sqsubseteq$, are also very important as they can be exploited for query optimization and for enforcing integrity constraints. This kind of issues are still unexplored and are subject of further research.

6 Concluding Remarks

This paper describes an approach for building P2P systems that support semantic-based retrieval services by extending the mediator model presented in [23]. The contents of the objects and the queries are expressed in terms of taxonomies. As each peer uses its own taxonomy for describing the contents of its objects and for formulating queries to the other peers, peers are equipped with inter-taxonomy mappings in order to carry out the required translation tasks. The adopted conceptual modeling approach (taxonomies, and inter-taxonomy mappings) has three main advantages: First, it is very easy to create the conceptual model of a source. Second, the integration of information from multiple sources can be done easily. Third, automatic articulation using data-driven methods is possible.

We gave the semantics for this kind of systems and identified the differences between query evaluation in mediators and query evaluation in P2P systems. Issues for further research include techniques for query optimization and for identification of the global relationships. Another important and very interesting issue for further research is the automatic or semi-automatic construction of articulations. Currently, we are investigating an ostensive data-driven method for automatic articulation [22].

References

1. S. Amba. "Automatic Linking of Thesauri". In *SIGIR'96*, Zurich, Switzerland, 1996.
2. 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.
3. 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 *Procs. of Measurement and Modeling of Computer Systems*, June 2000.
4. D. Calvanese, G. De Giacomo, and M. Lenzerini. A framework for ontology integration. In *SWWS'2001*, 2001.
5. A. Doan, J. Madhavan, P. Domingos, and A. Halevy. "Learning to Map between Ontologies on the Semantic Web". In *WWW-2002*, 2002.
6. V. Kashyap and A. Sheth. "Semantic Heterogeneity in Global Information Systems: the Role of Metadata, Context and Ontologies ". In *Cooperative Information Systems: Trends and Directions*. Academic Press, 1998.
7. 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.
8. M. Lenzerini. Data integration: A theoretical perspective. In *Proc. ACM PODS 2002*, Madison, Wisconsin, USA, June 2002.
9. A. Y. Levy. "Answering Queries Using Views: A Survey". *VLDB Journal*, 2001.
10. B. Ling, Z. Lu, W. Siong Ng, B. Ooi, Kian-Lee Tan, and A. Zhou. "A Content-Based Resource Location Mechanism in PeerIS". In *Procs. WISE'2002*, Singapore, Dec. 2002.
11. S. Luke, L. Spector, D. Rager, and J. Hendler. "Ontology-based Web Agents". In *Procs of 1st Int. Conf. on Autonomous Agents*, 1997.
12. P. Mitra, G. Wiederhold, and J. Jannink. "Semi-automatic Integration of Knowledge sources". In *Proc. of the 2nd Int. Conf. On Information FUSION*, 1999.
13. W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmer, and T. Risch. "EDUTELLA: A P2P networking infrastructure based on RDF". In *WWW'2002*, 2002.
14. R. Prieto-Diaz. "Implementing Faceted Classification for Software Reuse". *Communications of the ACM*, 34(5), 1991.
15. 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.
16. I. Ryutaro, T. Hideaki, and H. Shinichi. "Rule Induction for Concept Hierarchy Allignment". In *Procs. of the 2nd Workshop on Ontology Learning at the 17th Int. Conf. on AI (IJCAI)*, 2001.
17. G. M. Sacco. "Dynamic Taxonomies: A Model for Large Information Bases". *IEEE Transactions on Knowledge and Data Engineering*, 12(3), May 2000.

18. M. Sintichakis and P. Constantopoulos. “A Method for Monolingual Thesauri Merging”. In *SIGIR’97*, Philadelphia, PA, USA, July 1997.
19. N. Spyratos, Y. Tzitzikas, and V. Christophides. “On Personalizing the Catalogs of Web Portals”. In *FLAIRS’02*, Pensacola, Florida, May 2002.
20. I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. “Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications”. In *SIGCOMM’2001*, 2001.
21. Y. Tzitzikas, A. Analyti, N. Spyratos, and P. Constantopoulos. “An Algebraic Approach for Specifying Compound Terms in Faceted Taxonomies”. In *13th Europ.-Jap. Conf. on Information Modelling and Knowledge Bases*, Japan, June 2003.
22. Y. Tzitzikas and C. Meghini. “Ostensive Automatic Schema Mapping for Taxonomy-based Peer-to-Peer Systems”. In *CIA-2003*, Helsinki, Finland, August 2003.
23. Y. Tzitzikas, N. Spyratos, and P. Constantopoulos. “Mediators over Ontology-based Information Sources”. In *WISE’2001*, Kyoto, Japan, December 2001.
24. J. D. Ullman. *“Principles of Database and Knowledge-Base Systems, Vol. I”*. Computer Science Press, 1988.
25. F. v. Harmelen and D. Fensel. “Practical Knowledge Representation for the Web”. In *Workshop on Intelligent Information Integration, IJCAI’99*, 1999.