

Mediators over Ontology-based Information Sources

Yannis Tzitzikas^{1,2,*}

Nicolas Spyratos^{3, *}

Panos Constantopoulos^{1,2}

¹ Department of Computer Science, University of Crete, Greece

² Institute of Computer Science, ICS-FORTH

³ Laboratoire de Recherche en Informatique, Université de Paris-Sud, France

Email : tzitzik@ics.forth.gr, spyratos@lri.fr, panos@csi.forth.gr

Abstract

We propose a model for providing integrated and unified access to multiple information sources. Each information source comprises two parts: (a) an ontology i.e. a set of terms structured by a subsumption relation, and (b) a database that stores objects under the terms of the ontology. We assume that the objects of interest belong to an underlying domain that is common to all sources (e.g. a set of web pages of interest), and that different sources may use different ontologies with terms that correspond to different natural languages or to different levels of granularity. Information integration is obtained through a mediator comprising two parts: (a) an ontology, and (b) a set of articulations to the information sources. Here, by articulation to a source we mean a set of relationships between terms of the mediator and terms of that source. Information requests (queries) are addressed to the mediator whose task is to analyze each query into sub-queries, translate them into queries to the appropriate sources, then merge the results to answer the original query. We study the querying and answering process in such a model and present algorithms for handling the main tasks of the mediator, namely, query translation between the mediator and the sources, source selection and result merging to produce the final answer.

1. Introduction

The need for integrated and unified access to multiple information sources has stimulated the research on *mediators* (initially proposed in [23]). Roughly a mediator is a "secondary" information source aiming at providing a uniform interface to a number of underlying sources (which may be primary or secondary). Users submit queries expressed over the ontology of the mediator. Upon receiving a user query, the mediator has to query the underlying sources. This involves selecting the sources to be queried and formulating

the query to be sent to each source. These tasks are accomplished based on what the mediator "knows" about the underlying sources. Finally, the mediator has to appropriately combine the returned results and deliver the final answer to the user.

In this paper we consider information *sources* over a domain consisting of a denumerable set of objects. For example in the environment of the web, the domain could be the set of all web pages, specifically, the set of all pointers to web pages. Each source has an *ontology*, that is, a structured set of names, or *terms*, that are familiar to the users of the source. In particular the ontologies that we consider in this paper consist of a set of terms structured by a subsumption relation. In addition, each source maintains a database storing objects that are of interest to its users. Specifically, each object in the database of a source is indexed under one or more terms of the ontology of that source. A user who looks for objects of interest can browse the ontology of the source until he reaches the desired terms, or he can query the source by submitting a boolean expression of terms. The source will then return the appropriate set of objects. In the environment of the web there are many examples of such sources. Specifically, the general purpose catalogs of the web, such as Yahoo! or Open Directory ¹, the domain specific catalogs/gateways (e.g. for medicine, physics, tourism), as well as the personal bookmarks of the web browsers can be considered as examples of such sources.

However, although several sources may carry information about the same domain, they usually employ different ontologies, with terms that correspond to different natural languages, or to different levels of granularity, and so on. For example, consider two sources S_1 and S_2 that provide access to electronic products as shown in figures 1.(a) and 1.(b). Each source consists of an ontology plus a database that stores objects under the terms of that ontology. Suppose now that we want to provide a unified access to these

*Work conducted while these authors were visiting with Meme Media Laboratory, Hokkaido University, Sapporo, Japan

¹<http://dmoz.org>

two sources through a single ontology which is familiar to a specific group of users. An example of such a unifying ontology is shown in Figure 1.(c), and constitutes part of what we call a "mediator".

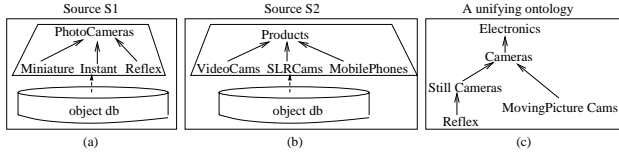


Figure 1. Two sources providing access to electronic products

A *mediator* is a secondary source that can bridge the heterogeneities that may exist between two or more sources and provides a unified access to those sources. Specifically, a mediator has an ontology with terminology and structuring that reflects the needs of its potential users, but does *not* maintain a database of objects. Instead, the mediator has 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 are defined by the designer of the mediator at design time and are stored at the mediator. Figure 2 shows the general architecture of a mediator.

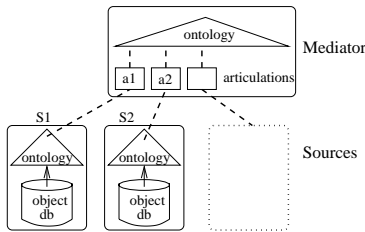


Figure 2. The mediator architecture

Users formulate queries over the ontology of the mediator and it is the task of the mediator to choose the sources to be queried, and the query to be sent to each source. Then it is again the mediator that appropriately combines the results returned by the sources in order to produce the final answer. Specifically, the mediator uses the articulations in order to translate queries over its own ontology to queries over the ontologies of the articulated sources.

An essential feature that distinguishes our approach is that the sources can provide two types of answer to a given query, namely, a *sure* answer or a *possible* answer (the first type of answer being appropriate for users that focus on precision, while the second for users that focus on recall). Moreover a user query to the mediator admits two types of translation, namely, *lower* or *upper* translation (again, the first type of translation being appropriate for users that focus on precision, while the second for users that focus on recall). What kind of translation will be used at the mediator level and what kind of answer will be requested at the source level

is decided by the mediator designer at design time and/or the mediator user at query time. Therefore a prominent feature of our approach is that sources and mediators can operate in a variety of modes according to specific application needs.

In the context of the web our mediators can be used for providing unified access to multiple web catalogs. An advantage of our model is that a mediator can be constructed quite easily, therefore ordinary web users can use our model in order to define their own mediators. In this sense, our model can be used for defining user views over existing web catalogs.

The remaining of the paper is organized as follows: Section 2 describes the information sources and the query answering process at a single source. Section 3 defines the architecture of a mediator and the different modes under which a mediator can operate. Section 4 discusses compatibility between the mediator and the sources, a condition that can lead to optimized query evaluation. Section 5 discusses related work and finally Section 6 concludes the paper and discusses further research.

2. The Information Sources

Let Obj denote the set of all objects of a domain (e.g. the set of all pointers to web pages). Each source has an *ontology*, i.e. a pair (T, \preceq) where T is a *terminology*, i.e. a set of names, or *terms*, and \preceq is a *subsumption* relation over T , i.e. a reflexive and transitive relation over T . If a and b are terms of T , we say that a is *subsumed* by b if $a \preceq b$; we also say that b *subsumes* a ; e.g. 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 T . Moreover \preceq is a partial order over the equivalence classes of terms.

In addition to its ontology, each source has a stored *interpretation* I of its terminology, i.e. a 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 power set of Obj . Figure 3 shows an example of a 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` as these objects are connected to `JournalArticle` with dotted arrows. Moreover as these are the only objects connected to `JournalArticle` with dotted arrows, they make up the interpretation of `JournalArticle`, i.e. $I(\text{JournalArticle}) = \{1, 3\}$. Subsumption of terms is indicated by a continuous-line arrow from the subsumed term to the subsuming term. For example, the term `RDB`

in Figure 3 is subsumed by DB as there is a continuous-line arrow going from RDB to DB, thus $RDB \preceq DB$. Note that we don't represent the whole subsumption relation but a subset of it sufficient to generate the whole relation. In particular, we don't represent the reflexive nor the transitive arrows of the subsumption relation. Equivalence of terms is indicated by a continuous non-oriented line, for example the term `Databases` is equivalent with `DB` since these terms are connected by a continuous non-oriented line.

For technical reasons that will become clear shortly, we assume that every terminology T contains two special terms, the *top term* denoted by \top and the *bottom term* denoted by \perp . The top term subsumes 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. $\perp \prec t$, for every t such that $t \neq \top$ and $t \neq \perp$. Moreover we assume that every interpretation I of T satisfies the condition $I(\perp) = \emptyset$.

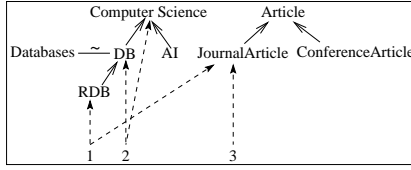


Figure 3. Graphical representation of a source

Concerning queries, each source responds to queries over its own terminology. Roughly, a query is either a term of a combination of terms using the usual connectives \wedge , \vee , \neg and $()$. For technical reasons that will become clear shortly we shall also use the concept of *empty query* denoted by ϵ . More formally, a query is defined as follows:

Def 2.1 Let T be a terminology. A *query* over T is any string derived by the following grammar, where t is a term of T :

$$q ::= t \mid q \wedge q' \mid q \vee q' \mid q \wedge \neg q' \mid (q) \mid \epsilon$$

Note that our use of negation corresponds to domain restricted negation.

Given two interpretations I, I' of T , we call I less than or equal to I' , and we write $I \sqsubseteq I'$, if $I(t) \subseteq I'(t)$ for each term $t \in T$. Note that \sqsubseteq is a partial order over interpretations.

A source 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 its ontology (i.e. the relation \preceq). For example, assume that a source has stored two sets of objects (e.g. two sets of pointers to web pages) under the terms `DB` and `AI`, and no objects under the term `Computer Science` - although the latter term subsumes the former two. However, such a stored interpretation is acceptable since we can "satisfy" the subsumption relation \preceq by defining the interpretation of `Computer Science` to be the union of the sets of objects indexed under `DB` and

`AI`. In order to define this formally we introduce the notion of *model*.

Def 2.2 An interpretation I is a model of an ontology (T, \preceq) if for each t, t' in T , if $t \preceq t'$ then $I(t) \subseteq I(t')$.

For brevity hereafter we shall also write T instead of (T, \preceq) , whenever no confusion is possible.

Now, as there may be several models of T in general, we assume that each source answers queries from one or more *designated* models of its stored interpretation. In this paper we will use two specific models for answering queries, the *sure model* and the *possible model*. In order to define these models formally we need a preliminary definition.

Def 2.3 Given a term $t \in T$ we define

$$tail(t) = \{s \in T \mid s \preceq t\} \text{ and } head(t) = \{u \in T \mid t \preceq u\}$$

Note that t , and all terms equivalent to t , belong to both $tail(t)$ and $head(t)$. Also note that $tail(t)$ always contains the bottom term and $head(t)$ always contain the top term.

Def 2.4 Given an interpretation I of T we define the *sure model* of T generated by I , denoted I^- , as follows:

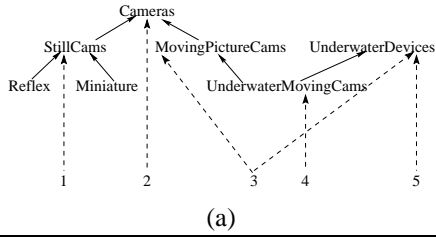
$$I^-(t) = \bigcup \{I(s) \mid s \in tail(t)\}$$

Intuitively the stored set $I(t)$ consists of the objects that are known to be indexed under t . The set $I^-(t)$ on the other hand consists of the objects known to be indexed under t plus the objects that are known to be indexed under terms subsumed by t . Therefore $I^-(t)$ consists of all objects that are *surely* indexed under t with respect to I and \preceq . Figure 4 shows an example of a source and its sure model I^- . Clearly I^- is a model of T and actually I^- is the unique minimal model greater than or equal to I .

Def 2.5 Given an ontology T and interpretation I we define the *possible model* of T generated by I , denoted I^+ , as follows:

$$I^+(t) = \bigcap \{I^-(u) \mid u \in head(t) \text{ and } u \not\approx t\}$$

As it is clear from its definition, the set $I^+(t)$ consists of the objects known to be indexed under each term strictly subsuming t . Therefore $I^+(t)$ consists of all objects that are *possibly* indexed under t with respect to I and \preceq . An example of the possible model of a source is given in Figure 4. Note that the possible interpretations of the terms `Cameras` and `UnderwaterDevices` is the set of *all* stored objects. This is so because the head of each of these terms contains only the term itself and the top term, thus we have: $I^+(\text{Cameras}) = I^+(\text{UnderwaterDevices}) = I^-(\top)$. Clearly I^+ is a model of T and we have $I \sqsubseteq I^- \sqsubseteq I^+$. It follows that for every term t we have $I^-(t) \subseteq I^+(t)$.



(a)

Term	I	I^-	I^+
\perp	\emptyset	\emptyset	\emptyset
\top	\emptyset	$\{1,2,3,4,5\}$	$\{1,2,3,4,5\}$
Cameras	$\{2\}$	$\{1,2,3,4\}$	$\{1,2,3,4,5\}$
StillCams	$\{1\}$	$\{1\}$	$\{1,2,3,4\}$
Reflex	\emptyset	\emptyset	$\{1\}$
Miniature	\emptyset	\emptyset	$\{1\}$
MovingPictureCams	$\{3\}$	$\{3,4\}$	$\{1,2,3,4\}$
UnderWaterMovingCams	$\{4\}$	$\{4\}$	$\{3,4\}$
UnderWaterDevices	$\{3,5\}$	$\{3,4,5\}$	$\{1,2,3,4,5\}$

(b)

Figure 4. Graphical representation of a source

In our approach we view the stored interpretation I as the result of indexing. However, although we may assume that indexing is done correctly, certain objects may not have been indexed under all terms that could apply to them. For example object 1 in Figure 4 is indexed under *StillCams* but not under *Cameras*, and object 3 is indexed under *MovingPictureCams* and *UnderwaterDevices* but not under *UnderwaterMovingCams* (although object 3 could in fact be an *UnderwaterMovingCamera*). As a consequence given a query that consists of a single term t we may want to answer it in either of two ways: (a) by including in the answer only objects that are known to be indexed under t , or (b) by including in the answer objects that are possibly indexed under t . In the first case the answer is the set $I^-(t)$, while in the second it is the set $I^+(t)$.

Referring to Def. 2.1 let us now define query answering for a general query q .

Def 2.6 Let q be a query over a terminology T and let I be an interpretation of T .

(a) The sure answer $I^-(q)$ is defined as follows:

$$\begin{aligned}
 I^-(t) &= \bigcup \{I(s) \mid s \in \text{tail}(t)\} \\
 I^-(q \wedge q') &= I^-(q) \cap I^-(q') \\
 I^-(q \vee q') &= I^-(q) \cup I^-(q') \\
 I^-(q \wedge \neg q') &= I^-(q) \setminus I^-(q') \\
 I^-(\epsilon) &= \emptyset
 \end{aligned}$$

(b) The possible answer $I^+(q)$ is defined as follows:

$$\begin{aligned}
 I^+(t) &= \bigcap \{I^-(u) \mid u \in \text{head}(t) \text{ and } u \not\sim t\} \\
 I^+(q \wedge q') &= I^+(q) \cap I^+(q') \\
 I^+(q \vee q') &= I^+(q) \cup I^+(q') \\
 I^+(q \wedge \neg q') &= I^+(q) \setminus I^-(q') \\
 I^+(\epsilon) &= \emptyset
 \end{aligned}$$

It follows from the above definition that for every query q we have $I^-(q) \subseteq I^+(q)$. This means that the sure answer of a query q is always included in the possible answer of q .

Note that we interpret $I^+(q \wedge \neg q')$ by $I^+(q) \setminus I^-(q')$, and *not* by $I^+(q) \setminus I^+(q')$. This is because if we interpreted $I^+(q \wedge \neg q')$ by $I^+(q) \setminus I^+(q')$ then we could find queries q for which $I^-(q) \supset I^+(q)$, contrary to intuition. For example, consider a terminology T with three terms a, b and c such that $c \preceq b \preceq a$, and an interpretation I such that $I(c) = \emptyset$, $I(b) = \{1\}$ and $I(a) = \{2\}$. Then for $q = a \wedge \neg c$ we have: $I^-(q) = I^-(a) \setminus I^-(c) = \{1, 2\}$ and $I^+(q) = I^+(a) \setminus I^+(c) = \{2\}$, i.e. $I^-(q) \supset I^+(q)$. However, with our definition we have $I^+(a \wedge \neg c) = I^+(a) \setminus I^-(c) = \{1, 2\}$, i.e. the relation $I^- \sqsubseteq I^+$ is preserved.

User interaction with the source consists in submitting a query q plus the nature of the desired answer (sure or possible). The system then responds by computing $I^-(q)$ or $I^+(q)$ according to the user's desire. We note that the possibility of providing two types of answer to a query can enhance the quality of user interaction with the source. For example the user may submit a query and require a sure answer. If the sure answer is empty this may mean either that no object has been indexed under the user's query or that the objects have been indexed to a coarser level. So, if the sure answer turns out to be empty the user can ask for the possible answer to his query. In the possible answer the user can see objects related to but not necessarily indexed under his query. Another possibility is that the sure answer to the query is not empty but the user just likes to see more objects related to his query, but at a coarser level. He can then ask for a possible answer to his query.

Concerning query evaluation at a source there are roughly two approaches. The first approach consists in computing and storing the models I^- and I^+ and then using these stored models for computing answers to queries. This can be done using algorithms that follow easily from Definition 2.6. The advantage of this approach is that answers can be computed in a straightforward manner from the stored models. The disadvantage is increased space requirements as well as increased maintenance costs for the stored models. Indeed, whenever the ontology or the interpretation I changes, I^- and I^+ must be updated appropriately. This requires an efficient method for handling updates since recomputing I^- and I^+ from scratch would be inefficient.

The second approach consists in storing only the interpretation I and, whenever a query q is received, computing the appropriate answer, $I^-(q)$ or $I^+(q)$. The computation of $I^-(q)$ can be done in a straightforward manner following Definition 2.6.(a). The computation of $I^+(q)$ can be done following again Definition 2.6.(b) but requires the previous computation of $I^-(t)$ for all terms t that subsume terms which appear in the query. The advantage of this approach is that we have no additional space requirements and no additional maintenance costs. The disadvantage is increased time cost for the computation of the answers.

Clearly the relative merits of the two approaches depend

on the application at hand as well as on the frequency by which the ontology and/or the stored interpretation of the source are updated. Note that in both approaches we need algorithms for computing the head and the tail of a term. However, if we compute the transitive closure of the subsumption relation, by one of the existing algorithms, then the algorithms for computing the head and tail of a term follow immediately from Definition 2.3.

3. The Mediator

We have seen so far that an information source over an underlying set Obj of objects consists of an ontology (T, \preceq) , and a stored interpretation I of T . The terminology T contains terms that are familiar to the users of the source; the subsumption relation \preceq contains relationships between terms as perceived by the users; and the stored interpretation I associates each term t with the objects that are indexed under t , by the indexer.

Consider now a set of sources S_1, \dots, S_k over the *same* underlying set Obj of objects. In general, two different sources may have different terminologies either because the users of the two sources are familiar with different sets of terms, or because one source classifies objects at a different level of granularity than the other. The two sources may also have different subsumption relations as the relationships between any two given terms may be perceived differently in the two sources. Finally, two different sources may have different stored interpretations, for example some objects may be indexed by one source but not by the other.

Clearly if one wants to combine or *integrate* information coming from different sources one has to cope with the above heterogeneities. In this paper we propose the use of *mediators* as a means for rendering all these heterogeneities transparent to users.

In our approach, a mediator M has an ontology (T, \preceq) that reflects the needs of its potential users but has *no* stored objects. Instead, each term at the mediator is related directly or indirectly with the terms in the underlying sources. More formally, a mediator is defined as follows:

Def 3.1 A mediator over k sources $S_1 = (T_1, \preceq_1), \dots, S_k = (T_k, \preceq_k)$ consists of:

- 1) an ontology (T, \preceq) , and
- 2) a set of *articulations* a_i , one for each source S_i ; each articulation a_i is a subsumption relation over $T \cup T_i$

Roughly speaking, a mediator is just like a source but with an important difference: there is no interpretation stored at the mediator. What is stored at the mediator, instead, is the set of articulations a_i , one for each source S_i . For example, suppose that we want to integrate two web catalogs which provide access to pages about electronic products. In particular consider the sources S_1 and S_2 shown in Figure 5

and assume that we want to provide access to these sources through a mediator M as shown in that figure. For achieving integration we enrich the mediator with articulations, i.e. with relationships that relate the terms of the mediator with the terms of the sources as shown in Figure 5. The articulations a_1 and a_2 are the following sets of subsumption relationships:

$$\begin{aligned}
 a_1 &= \{ \text{PhotoCameras} \preceq \text{Cameras}, \text{StillCameras} \preceq \text{PhotoCameras}, \\
 &\quad \text{Miniature} \preceq \text{StillCameras}, \\
 &\quad \text{Instant} \preceq \text{StillCameras}, \text{Reflex}_1 \preceq \text{StillCameras}, \\
 &\quad \text{Reflex}_1 \preceq \text{Reflex}, \text{Reflex} \preceq \text{Reflex}_1 \} \\
 a_2 &= \{ \text{Products} \preceq \text{Electronics}, \text{SLRCams} \preceq \text{Reflex}, \\
 &\quad \text{VideoCams} \preceq \text{MovingPictureCams}, \\
 &\quad \text{MovingPictureCams} \preceq \text{VideoCams} \}
 \end{aligned}$$

Note that a_1 is a subsumption relation over $T \cup T_1$ and a_2 is a subsumption relation over $T \cup T_2$, as required by the definition of an articulation (Def. 3.1).

Figure 6 shows another example of a mediator over three sources. These three sources provide access to tourist information and the information is organized by location.

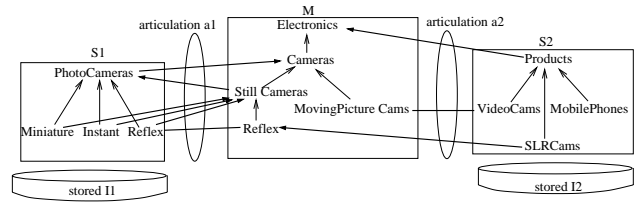


Figure 5. A mediator over two catalogs of electronic products

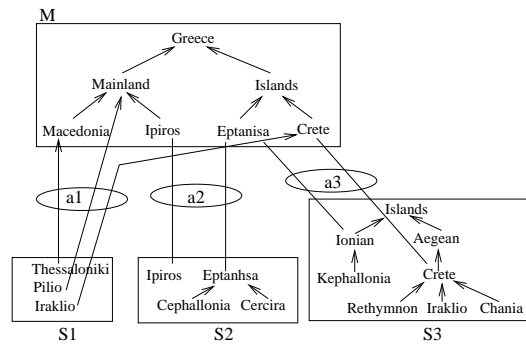


Figure 6. A mediator over three catalogs of tourist information

Now, in the presence of several sources, one and the same term may appear in two or more sources. If the same term appears in two different sources then we consider the two appearances as two different terms. This is implemented by subscripting each term of a source S_i by the subscript i . Take for example the term DB and suppose that it appears

in sources S_i and S_j . Then, from the mediator's point of view there are two terms: the term DB_i in source S_i and the term DB_j in source S_j . This is reasonable as the same term can have different interpretations (meanings) in different sources. Thus for every $i \neq j$ we assume $T_i \cap T_j = \emptyset$; and for every i we assume $T \cap T_i = \emptyset$. In this way we overcome the problems of homonyms. Under these assumptions, two terms are considered equivalent, e.g. $DB_i \sim DB_j$, only if they can be shown to be equivalent using the articulations a_i and a_j , e.g. DB_i and DB_j are equivalent if there is a term t in T such that $t \sim_{a_i} DB_i$ and $t \sim_{a_j} DB_j$.

Now, if the mediator is to answer user queries, we must define an interpretation I of its terminology, based on the interpretations I_i stored at the sources, on the one hand, and on the articulations a_i , $i = 1, \dots, k$, on the other hand. Once the interpretation I of the mediator is defined, the mediator will be able to answer queries just like any other source does, i.e. from its sure model I^- and from its possible model I^+ . In our approach, in order to define the mediator interpretation I we proceed as follows: for every term t of the mediator terminology T :

1. first we define an approximation t^i of t in a_i , in the form of a query at source S_i , $i = 1, \dots, k$;
2. then we evaluate the query t^i at source S_i , $i = 1, \dots, k$,
3. finally we define $I(t)$ by taking the union of the answers to the queries t^i returned by the sources.

However, there are two ways to approximate t using the articulation a_i , that we shall call the *upper approximation* of t and the *lower approximation* of t in a_i . Roughly, the upper approximation of t in a_i is the conjunction of all terms of T_i that subsume t in a_i , and the lower approximation of t in a_i is the disjunction of all terms of T_i that t subsumes in a_i . In order to define these notions formally we need a preliminary definition:

Def 3.2 Given a term $t \in T$ and articulation a_i we define

$$tail_i(t) = \{s \in T_i \mid sa_i t\} \text{ and } head_i(t) = \{u \in T_i \mid ta_i u\}$$

Def 3.3 Let $M = (T, \preceq, a_1, \dots, a_k)$ be a mediator over sources S_1, \dots, S_k . If t is a term of T then

- the *lower approximation* of t with respect to a_i , denoted t_l^i , is defined by
$$t_l^i = \bigvee tail_i(t)$$
- the *upper approximation* of t with respect to a_i , denoted t_u^i , is defined by

$$t_u^i = \begin{cases} \bigwedge head_i(t), & \text{if } head_i(t) \neq \emptyset \\ t_l^i, & \text{otherwise} \end{cases}$$

Here are some examples of approximations for the mediator shown in Figure 5:

```

StillCameras_l^1 = Miniature \vee Instant \vee Reflex
StillCameras_u^1 = PhotoCameras
Reflex_l^1 = Reflex
Reflex_u^1 = Reflex \wedge PhotoCameras
Reflex_l^2 = SLRCams
Cameras_l^1 = PhotoCameras \vee Miniature \vee Instant \vee Reflex
Cameras_u^1 = PhotoCameras \vee Miniature \vee Instant \vee Reflex
MovingPictureCams_u^1 = MovingPictureCams_l^1 = \epsilon

```

Note that for a given term $t \in T$ the evaluation of t_u^i requires the previous evaluation of $head_i(t)$, and the evaluation of t_l^i requires the previous evaluation of $tail_i(t)$. If we compute the transitive closure of a_i then the evaluation of $head_i(t)$ and $tail_i(t)$ is straightforward.

Now, the approximations t_u^i and t_l^i of t are actually queries to the source S_i , and as such each can have a sure answer and a possible answer (see Section 2). As a consequence, we can define at least four different interpretations I for the mediator. Assuming for simplicity that *all* sources respond in the same manner, i.e. either all give a sure answer or all give a possible answer, we can define exactly four interpretations for the mediator that we shall denote by I_{l-} , I_{l+} , I_{u-} , I_{u+} . These interpretations are defined as follows:

- 1 Lower approximation of t at mediator and sure answer from sources:

$$I_{l-}(t) = \bigcup_{i=1}^k I_i^-(t_l^i)$$

- 2 Lower approximation of t at mediator and possible answer from sources:

$$I_{l+}(t) = \bigcup_{i=1}^k I_i^+(t_l^i)$$

- 3 Upper approximation of t at mediator and sure answer from sources:

$$I_{u-}(t) = \bigcup_{i=1}^k I_i^-(t_u^i)$$

- 4 Upper approximation of t at mediator and possible answer from sources:

$$I_{u+}(t) = \bigcup_{i=1}^k I_i^+(t_u^i)$$

So, the mediator can answer queries submitted by its users based on any of the above four interpretations. Moreover, for any of these four interpretations, the mediator can give either a sure answer or a possible answer - just as any source can (see Section 2). As a consequence, we can distinguish eight possible modes under which a mediator can operate. Each mode essentially corresponds to a different answer model of the mediator. The operation modes of a mediator and the corresponding models are summarized in Table 1.

Very roughly speaking, as we go down the table (from mode 1 to 8) the answer to the same user query is more likely to contain objects that are not "relevant" to the query. This is described more precisely in Figure 7. The nodes represent the answer models shown in Table 1; for example, n_1 represents I_{l-} , n_2 represents I_{l+} , and so on. An arrow from node n_i to a node n_j means that $n_i \sqsubseteq n_j$.

An example of mediator operation is given in Figure 8. Figure 8.(a) shows a mediator having an articulation

operation mode at the mediator	term approx. at med.	query evaluation at source	query evaluation at med.	the answer model of the med.
1	lower	sure	sure	I_{l-}^-
2	lower	possible	sure	I_{l+}^-
3	upper	sure	sure	I_{u-}^-
4	upper	possible	sure	I_{u+}^-
5	lower	sure	possible	I_{l-}^+
6	lower	possible	possible	I_{l+}^+
7	upper	sure	possible	I_{u-}^+
8	upper	possible	possible	I_{u+}^+

Table 1. Scenarios under which a mediator can operate

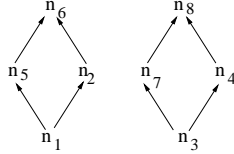


Figure 7. The ordering (\sqsubseteq) of the eight answer models of the mediator

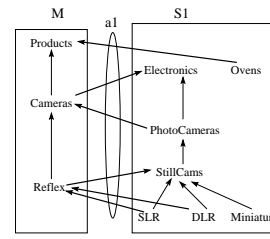
to a source S_1 and Figure 8.(b) shows two tables. The table at the upper part of the figure shows the interpretation I_1 of source S_1 and the corresponding (sure and possible) models. The first column of the table at the bottom part shows three queries which are actually the three terms of T_1 . The subsequent columns show what the mediator returns in each of the first four operation modes.

The operation modes of the mediator either can be decided (and fixed) by the mediator designer at design time, or they may be indicated by the mediator users at query time. We can distinguish at least three approaches:

- *Fixed Mode.* The mediator designer selects and fixes one of the eight possible modes of operation for the mediator and the sources, and users simply submit their queries to the mediator without any further indication.
- *User-centric Mode.* The mediator users submit their queries along with a specification for the query evaluation mode they wish. This is done by providing values to the mediator for selecting one of the eight operation modes from Table 1. For example, the following user specification selects the operation mode number 3 from Table 1:

term approximation at mediator = upper
query evaluation at source = sure
query evaluation at mediator = sure

- *Mixed Mode.* The mediator designer selects and fixes some of the attributes of Table 1, and the user provides the remaining ones. For example, the designer may



(a)

T_1	I_1	I_1^-	I_1^+
\perp	\emptyset	\emptyset	\emptyset
\top	\emptyset	$\{0,1,2,3,4,5,6\}$	$\{0,1,2,3,4,5,6\}$
Ovens	$\{6\}$	$\{6\}$	$\{0,1,2,3,4,5,6\}$
Electronics	$\{5\}$	$\{0,1,2,3,4,5\}$	$\{0,1,2,3,4,5,6\}$
PhotoCameras	$\{4\}$	$\{0,1,2,3,4\}$	$\{0,1,2,3,4,5\}$
StillCams	$\{3\}$	$\{0,1,2,3\}$	$\{0,1,2,3,4\}$
SLR	$\{2\}$	$\{2\}$	$\{0,1,2,3\}$
DLR	$\{1\}$	$\{1\}$	$\{0,1,2,3\}$
Miniature	$\{0\}$	$\{0\}$	$\{0,1,2,3\}$

Q	1: I_{l-}^-	2: I_{l+}^-	3: I_{u-}^-	4: I_{u+}^-
Products	$\{0,1,2,3,4,6\}$	$\{0,1,2,3,4,5,6\}$	$\{0,1,2,3,4,5,6\}$	$\{0,1,2,3,4,5,6\}$
Cameras	$\{0,1,2,3,4\}$	$\{0,1,2,3,4,5\}$	$\{0,1,2,3,4,5\}$	$\{0,1,2,3,4,5,6\}$
Reflex	$\{1,2\}$	$\{0,1,2,3\}$	$\{0,1,2,3\}$	$\{0,1,2,3,4\}$

(b)

Figure 8. A mediator with one one articulation to a source S_1

select and fix the query evaluation at source (i.e. sure or possible) and the term approximation at the mediator (i.e. lower or upper), during design time, while the users select the query evaluation mode at the mediator, during query time.

Clearly, selecting one of the above approaches depends on several factors, such as the reliability of the sources or the level of expertise of the users, and so on.

4. The Compatibility Condition

We have seen so far how the mediator communicates with the sources through the articulations. In fact, the articulations are the *only* means of communication between the sources and the mediator. Now, certain kinds of articulation are better than others. One kind of articulations that are of interest in this paper are those that ensure what we call "compatibility" between the sources and the mediator.

Def 4.1 A source S_i is *compatible* with the mediator M if for any terms s, t in T_i , if sa_it then $s \preceq_i t$.

That is, S_i is compatible with the mediator whenever the following condition holds: for all terms s and t in T_i , if s is subsumed by t in the articulation a_i then s is also subsumed by t in \preceq_i .

For example, the source S_1 of Figure 5 is compatible with the mediator since we have

Miniature a_1 PhotoCameras and
 Miniature \preceq_1 PhotoCameras,
 Instant a_1 PhotoCameras and
 Instant \preceq_1 PhotoCameras,
 Reflex a_1 PhotoCameras and
 Reflex \preceq_1 PhotoCameras.

An interesting consequence of compatibility is that if a source S_i is compatible with the mediator, then in every model I_i of S_i the following condition holds: the lower approximation of a term t is a subset of the upper approximation of t , that is, $I_i(t_l^i) \subseteq I_i(t_u^i)$, for each mediator term t . From this property we infer that if all sources are compatible with the mediator then the ordering relation over the eight answer models of the mediator (see Figure 7), is now enriched. As a result, the two diagrams of Figure 7 are now connected in a single diagram as shown in Figure 9.

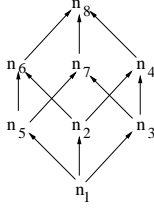


Figure 9. The ordering (\sqsubseteq) of the eight answer models of the mediator in the case where all sources are compatible

Note that the above ordering relationships do not hold if the sources are not compatible with the mediator. For example, consider a source S_1 with terminology $T_1 = \{b, b'\}$ and no subsumption relationships. Suppose that the source has a stored interpretation I_1 defined as follows: $I_1(b) = \{1\}$ and $I_1(b') = \{2\}$. Now consider a mediator connected to source S_1 through the articulation $a_1 = \{b \preceq t, t \preceq b'\}$, where t is a term of the mediator. Notice that S_1 is not compatible with the mediator because b is subsumed by b' in a_1 while b is not subsumed by b' in \preceq_1 , i.e. ba_1b' and $b \not\preceq_1 b'$. Here we have $t_l^1 = b$ and $t_u^1 = b'$, thus $I_1^-(t_l^1) = \{1\}$ and $I_1^-(t_u^1) = \{2\}$. It follows that $I_1^-(t_l^1) \not\subseteq I_1^-(t_u^1)$, which implies $I_1^-(t) \not\subseteq I_{u-}^-(t)$. From this example we see that if the underlying sources are not compatible with the mediator then it does not hold that $I_{l-}^- \subseteq I_{u-}^-$.

Let us now describe another implication of compatibility. Let s, t be two terms in T_i which are known to the mediator (through a_i) and assume that the mediator knows that source S_i is compatible. In this case if sa_it then $s \preceq_i t$. From this knowledge the mediator can conclude that $I_i(s) \subseteq I_i(t)$, in every model I_i of T_i , and thus $I_i(s) \cap I_i(t) = I_i(s)$ and $I_i(s) \cup I_i(t) = I_i(t)$. This means that the mediator can obtain only the minimal elements of the set $head_i(t)$ and still obtain the same answer for the query t_u^i from source S_i . Therefore, if the mediator knows that source S_i is compatible,

then instead of sending to source S_i the query $\bigwedge head_i(t)$, the mediator can send the query $\bigwedge min(head_i(t))$. Similarly, in the set $tail_i(t)$ the mediator can retain only the maximal elements and still obtain the same answer for the query t_l^i from source S_i , i.e., instead of sending the query $\bigvee tail_i(t)$ to source S_i , the mediator can send the query $\bigvee max(tail_i(t))$.

For example, in Figure 5, as source S_1 is compatible with the mediator, the lower approximation of the term Camera is now the term PhotoCameras. If S_1 were not compatible then the lower approximation of Camera would be the disjunction PhotoCameras \vee Miniature \vee Instant \vee Reflex.

Thus if S_i is compatible then $t_u^i = \bigwedge min(head_i(t))$ and $t_l^i = \bigvee max(tail_i(t))$. In this case the evaluation of t_u^i and t_l^i can be done more efficiently without having to compute the transitive closure of a_i . Specifically, for evaluating $max(tail_i(t))$ we traverse in depth-first-search the relation a_i starting from the term t . If an element t' of T_i is reached then this term is "collected" and the algorithm does not traverse any other element subsumed by t' (in a_i). All elements of T_i which were collected during the traversal are then returned. Analogously we can evaluate $min(head_i(t))$. We conclude that if a source is compatible then the approximation of a term for that source can be done more efficiently especially when the articulation to that source is big. Moreover the resulting approximations are shorter which implies that their transmission requires less time and that the underlying source can evaluate these queries more efficiently.

Note that maintaining compatibility is not an easy task. Of course, the designer of the mediator can initially design articulations such that the underlying sources are compatible. However, an update at a source S_i or at the mediator (changing either T or a_i) may destroy compatibility. For this purpose the mediator should (periodically) check the compatibility of its sources, e.g. by submitting to them queries allowing to check whether $t \preceq_i t'$.

5. Related Work

The need for integrated and unified access to multiple information sources has stimulated the research on *mediators*. The concept of mediator was initially proposed by Wiederhold [23]. Since then many different approaches have been proposed in order to build mediators over relational databases (e.g. see [13, 7, 8, 24]), SGML documents (e.g. see [5]), or information retrieval systems (e.g. see [22, 9, 6, 19, 16]) and web-based sources (e.g. see [1, 3]).

Comparing to the integration approaches for relational databases, namely, federated databases, relational warehousing, and relational mediation, our model is similar in spirit to the relational mediators (see [8] for a review).

Roughly, relational mediators operate similarly to our mediators, but there are some critical differences:

- Relational mediators and their sources are schema-based while our mediators and their sources are ontology-based.
- Relational mediators try to construct exact translations of SQL queries through wrappers (query templates) while our mediators allow approximate translations of boolean expressions through their articulations.
- Relational mediators are "rigid" as they offer a single mode of operation while our mediators are "flexible" as they allow multiple modes of operation.

The techniques for building relational mediators are appropriate for rendering the structural (schema) heterogeneities of the sources transparent to the users (see the systems TSIMMIS [4], [7], HERMES [18], Information Manifold [13]). However these approaches do not support approximate translation although there are many scenarios in which this functionality is necessary.

One approach that considers approximate translations is [3]. The queries considered there are boolean expressions of constraints of the form $[attr1 \text{ op } value]$ or $[attr1 \text{ op } attr2]$ and mapping rules are employed in order to handle differences in operators, data formats and attribute names. The translated queries minimally subsume the original ones. However the functionality offered by our mediators is different, firstly because we support negation while they do not, and secondly because our mediators support multiple operation modes, one of which is the case where the translated queries subsume the original ones. A different approach to mediators can be found in [2] which presents the fundamental features of a declarative approach to information integration based on Description Logics. The authors describe a methodology for integrating relational sources and they resort to very expressive logics in order to bridge the heterogeneities between the unified view of the mediator and the source views. However the reasoning services for supporting the needed translations have exponential complexity as opposed to the complexity of our mediators which is clearly polynomial (see the full paper [21] for more details). Another relevant system aiming at translating queries over multiple distributed and heterogeneous ontology-based sources is the system OBSERVER ([15], [12]). It employs interontology relations and the translation of queries is based on intentional and extensional properties. The difference with our work is that their approach requires merging the ontologies of all underlying sources. Instead, we just articulate the ontologies of the sources with the ontology of the mediator. Moreover we have introduced the compatibility condition which allows the mediator to draw conclusions about the structuring of a source ontology without having to store that ontology. As for the articulations that we consider, they can be defined by humans, but they can also be constructed

automatically or semi-automatically in some specific cases. For example, there is a statistical method for constructing articulations based on the analysis of co-occurrence of terms within a parallel corpus [11]. However such methods require having a big number of objects that are indexed by multiple ontologies - something very difficult to have.

6. Concluding Remarks

We have seen an approach for providing uniform access to multiple information sources through mediators that render the heterogeneities of the sources transparent to users. A prominent feature of our approach is that a mediator is seen as just another source but *without* stored interpretation. An interpretation for the mediator is defined based on the interpretations stored at the sources and on the articulations between the mediator and the sources; and in fact, we have seen eight different ways for defining a mediator interpretation depending on the nature of the answers that the mediator provides to its users.

An important advantage of our approach is that we can create a complex information network, comprising sources and mediators, in a natural and straightforward manner. Indeed, in order to add a mediator to such a network one has to (a) select the sources to be mediated, (b) design the mediator ontology (T, \preceq) based on the ontologies (T_i, \preceq_i) of the selected sources, (c) design the articulations a_i based on the known/observed relations between terms of the mediator and terms at the selected sources. And to remove a mediator from the information network one has to just disconnect the mediator from the network. Similarly, in order to add a source to the information network one has to select a mediator in the network and design an articulation between the selected mediator and the new source.

As a result, we believe that our approach provides a flexible and formal framework for constructing mediators over ontology-based information sources, in particular, over web-based information sources where the objects of interest are usually just pointers to web pages. The ontologies that we consider fit quite well with the content-based organizational structure of web catalogs and portals (e.g. Yahoo!, Open Directory²), keyword hierarchies (e.g. ACM's thesaurus) and personal bookmarks. Besides most of the ontologies that are used for indexing and retrieving objects are term hierarchies ([10], [14], [17]). Since many ontologies (e.g. those employed by web catalogs) usually contain very large numbers of terms, the articulation of ontologies has many advantages comparing to ontology merging. Clearly, merging the ontologies of all underlying sources would introduce storage and performance overheads. In addition, full integration is a laborious task which in many cases does not pay-off because the integrated ontology becomes obsolete when the

²<http://dmoz.org>

involved ontologies change. Another advantage of ontology articulation (instead of merging) is that merging in general requires full consistency which may be hard to achieve in practice, while articulation can work on locally consistent parts of the involved ontologies. However notice that for the ontologies considered in this paper, we have no consistency problems (we can only have long cycles of \preceq -relationships, which induce big classes of equivalent terms).

Finally we believe that our approach renders itself to rapid prototyping - a task that admittedly we have not yet addressed. In this paper we have focused mainly on the definition of a formal framework where mediation can be accomplished through articulations. However, prototyping and experimenting with real data especially in the context of web sources are the immediate topics in our research agenda.

Acknowledgements. Many thanks to Professor Tanaka, director of the Meme Media Laboratory, and to the University of Hokkaido for their hospitality, and to Anastasia Analyti for her editorial comments.

References

- [1] J. L. Ambite, N. Ashish, G. Barish, C. A. Knoblock, S. Minton, P. J. Modi, I. Muslea, A. Philpot, and S. Tejada. Ariadne: a system for constructing mediators for Internet sources. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 561–563, 1998.
- [2] D. Calvanese, G. de Giacomo, M. Lenzerini, D. Nardi, and R. Rosati. "Description Logic Framework for Information Integration". In *Proceedings of the 6th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR-98)*, 1998.
- [3] C.-C. K. Chang and H. García-Molina. "Mind Your Vocabulary: Query Mapping Across Heterogeneous Information Sources". In *Proc. of the ACM SIGMOD*, pages 335–346, 1999.
- [4] S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, and J. Widom. "The TSIMMIS project: Integration of Heterogeneous Information Sources". In *Proceedings of IPSJ*, Tokyo, Japan, October 1994.
- [5] S. Cluet, C. Delobel, J. Siméon, and K. Smaga. "Your mediators need data conversion!". In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 1998.
- [6] N. Fuhr. "A Decision-Theoretic Approach to Database Selection in Networked IR". *ACM Transactions on Information Systems*, 17(3), July 1999.
- [7] H. Garcia-Molina, Y. Papakonstantinou, D. Quass, A. Rajaraman, Y. Sagiv, J. Ullman, V. Vassalos, and J. Widom. "The TSIMMIS Approach to Mediation: Data Models and Languages". In *Proceedings of IPSJ*, Tokyo, Japan, October 1994.
- [8] H. Garcia-Molina, J. D. Ullman, and J. Widom. "Database System Implementation", chapter 11. Prentice Hall, 2000.
- [9] L. Gravano and H. Garcia-Molina. "Generalizing GLOSS To Vector-Space Databases and Broker Hierarchies". In *Proc 21st VLDB Conf.*, Zurich, Switzerland, 1996.
- [10] N. Guarino. "Formal Ontology and Information Systems". In *Proceedings of FOIS'98*, Trento, Italy, June 1998. Amsterdam, IOS Press.
- [11] H. Helleg, J. Krause, T. Mandl, J. Marx, M. Muller, P. Mutschke, and R. Strogon. "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/).
- [12] 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.
- [13] A. Y. Levy, D. Srivastava, and T. Kirk. "Data Model and Query Evaluation in Global Information Systems". *Journal of Intelligent Information Systems*, 5(2), 1995.
- [14] D. L. McGuinness. "Ontological Issues for Knowledge-Enhanced Search". In *Proceedings of FOIS'98*, Trento, Italy, June 1998. Amsterdam, IOS Press.
- [15] E. Mena, V. Kashyap, A. Sheth, and A. Illarramendi. "OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperation across Preexisting Ontologies.". In *Proceedings of the First IFCIS International Conference on Cooperative Information Systems (CoopIS'96)*, Brussels, Belgium, June 1996. IEEE Computer Society Press.
- [16] H. Nottelmann and N. Fuhr. "MIND: An Architecture for Multimedia Information Retrieval in Federated Digital Libraries". In *DELOS Workshop on Interoperability in Digital Libraries*, Darmstadt, Germany, September 2001.
- [17] A. Pretschner. "Ontology Based Personalized Search". Master's thesis, Department of Electrical Engineering and Computer Science - University of Kansas, 1999.
- [18] V. S. Subrahmanian, S. Adah, A. Brink, R. Emery, A. Rajput, R. Ross, T. Rogers, and C. Ward. "HERMES: A Heterogeneous Reasoning and Mediator System", 1996. (www.cs.umd.edu/projects/hermes/overview/paper).
- [19] Y. Tzitzikas. "Democratic Data Fusion for Information Retrieval Mediators". In *ACS/IEEE International Conference on Computer Systems and Applications*, Beirut, Lebanon, June 2001.
- [20] Y. Tzitzikas, N. Spyrtos, and P. Constantopoulos. "Deriving Valid Expressions from Ontology Definitions". In *11th European-Japanese Conference on Information Modelling and Knowledge Bases*, Maribor, Slovenia, May 2001.
- [21] Y. Tzitzikas, N. Spyrtos, and P. Constantopoulos. "Mediators over Ontology-based Information Sources", 2001. (submitted for publication).
- [22] E. Vorhees, N. Gupta, and B. Johnson-Laird. "The Collection Fusion Problem". In *Proceedings of the Third Text Retrieval Conference (TREC-3)*, Gaithersburg, MD, 1995.
- [23] G. Wiederhold. "Mediators in the Architecture of Future Information Systems". *IEEE Computer*, 25:38–49, 1992.
- [24] R. Yerneni, C. Li, H. Garcia-Molina, and J. Ullman. "Computing capabilities of mediators". In *Proceedings of ACM SIGMOD'99*, Philadelphia, 1999.