

Ontology Matching Supported by Query Answering in a P2P System

François-Élie Calvier, Chantal Reynaud

► **To cite this version:**

François-Élie Calvier, Chantal Reynaud. Ontology Matching Supported by Query Answering in a P2P System. 7th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE 2008), Nov 2008, Monterrey, Mexico. pp.1559-1567, 10.1007/978-3-540-88873-4 . hal-00530142

HAL Id: hal-00530142

<https://hal.archives-ouvertes.fr/hal-00530142>

Submitted on 28 Oct 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Ontology Matching Supported by Query Answering in a P2P System

François-Élie Calvier and Chantal Reynaud

LRI, Univ Paris-Sud & INRIA Saclay - Île-de-France
4, rue Jacques Monod - Bât. G
91893 Orsay Cedex FRANCE
{francois.calvier, chantal.reynaud}@lri.fr
<http://www.lri.fr/iasi>

Abstract. In this article we propose methods to automate the extraction of alignments or mappings between ontologies by using query answering in the peer data management system (PDMS) SomeRDFS which uses a data model based on RDF(S). Query answering is composed of a rewriting and an evaluation phase. We show that query answering provides information that offer automated support for discovering new mappings. It is used to generate either mapping shortcuts corresponding to mapping compositions or mappings which can not be inferred from the network but yet relevant. The strategy followed by a peer and filtering criteria defined by the administrator of a peer are used to select the most relevant mappings to be represented.

Key words: ontology matching, peer-to-peer, data management systems.

1 Introduction

The use of peer-to-peer systems consists of querying a network of peers for information. Queries are asked to one of the peers. The peers communicate to each other to answer queries in a collective way.

We focus on the ontology matching process in the peer data management system (PDMS) SomeRDFS [1] in the setting of the MediaD project¹. Ontologies are the description of peers data. Peers in SomeRDFS interconnect through alignments or mappings which are semantic correspondences between their own ontologies. Thanks to its mappings a peer may interact with the others in order to answer a query. No peer has a global view of the data management system. Each peer has its own ontology, its own mappings and its own data. It ignores the ontology, the mappings and the data of the other peers. In this setting, our work aims at increasing the mappings of the peers in SomeRDFS in order to increase the quantity and the quality of the answers of the whole data management system. We are interested in identifying two kinds of mappings: mapping shortcuts corresponding to a composition of pre-existent mappings and mappings which can not be inferred from the network but yet relevant. In both cases, the idea is to make the generation of mappings automatically supported by query answering. We

¹ Research project funded by France Telecom R&D

take also into account the strategy followed by a peer which is important to select the most relevant mappings to be represented.

A survey of usual methods for automating the generation of mappings is presented in [2], [3] and [4] but very little work has been done on this problem in a P2P environment taking into account its distinguishing features, in particular the distributed global schema or ontology. For example, in Piazza [5], tools and techniques developed to simplify and assist mapping creation [6] assume that the whole ontology of a peer can be known by any other peer. In [7], shared ontologies are used. Unlike these approaches we propose techniques to generate mappings between entirely decentralized ontologies. Each peer has its own ontology and ignores the ontology of the others. Furthermore, queries are not specific and are not routed to all peers over the network as in [8]. They are usual queries submitted to SomeRDFS and usual SomeRDFS reasoning mechanisms are reused. These mechanisms are exploited in two ways. We generate mapping shortcuts corresponding to mapping compositions. These mappings are produced from the routing of the queries over the network. The problem we are interested in is then to select the mappings useful to be represented. This differs from works whose goal is to produce mapping composition algorithms [9]. Furthermore, relevant elements to be mapped are selected. They share a common interpretation context making the alignment process easier by avoiding misinterpretations. The identification of context used for focusing the matching process is also a solution provided in [10] based on the analysis of the interactions between agents that are assumed to follow conventions and pattern. We share this idea but differ in its accomplishment. Finally, our work can be seen as a complement of [11] whose goal is to identify methods to establish global forms of agreement from a graph of local mappings among schemas. This work assumes that skilled experts supported by appropriate mapping tools provide the mappings. Our approach provides automated techniques to generate these mappings.

The paper is organized as follows. Section 2 describes the fragment of RDF(S) that we consider as data model for SomeRDFS and query answering. Section 3 shows how the query answering process can be used to discover new mappings according to a given strategy of a peer. We conclude and outline remaining research issues in Section 4.

2 Data Model and Query Answering in a SomeRDFS PDMS

In SomeRDFS ontologies and mappings are expressed in RDF(S) and data are represented in RDF. (Sub)classes, (sub)properties can be defined. Domain and range of properties can be typed. Classes inclusion, properties inclusion, domain and range typing of a property are the only authorized constructors. This language, denoted core-RDFS, has a clear and intuitive semantics. It is constructed on unary relations that represent classes and binary relations that represent properties. The logical semantics of core-RDFS, expressed in description logic (DL notation) and its translation in first-order logic (FOL), is given in Table 1.

Peers ontologies are made of core-RDFS statements involving the vocabulary of only one peer. We use the notation $\mathcal{P}:R$ for identifying the relation (class or property) R of the ontology of the peer \mathcal{P} . A mapping is an inclusion statement between classes or properties of two distinct peers (cf. Table 2 (a) and (b)) or a typing statement of a

Table 1. Core-RDF(S) operators

Operator	DL Notation	FOL translation
Class inclusion	$C_1 \sqsubseteq C_2$	$\forall X, C_1(X) \Rightarrow C_2(X)$
Property inclusion	$P_1 \sqsubseteq P_2$	$\forall X \forall Y R_1(X, Y) \Rightarrow R_2(X, Y)$
Domain typing of a property	$\exists P \sqsubseteq C$	$\forall X \forall Y, P(X, Y) \Rightarrow C(X)$
Range typing of a property	$\exists P^- \sqsubseteq C$	$\forall X \forall Y, P(X, Y) \Rightarrow C(Y)$

property of a given peer with a class of another peer (cf. Table 2 (c) and (d)). Mappings are defined as core-RDFS statements involving vocabularies of different peers.

The specification of the data stored in a peer is done through the declaration of assertion statements relating data of a peer to relations of its vocabulary. The DL notation and the FOL translation of assertion statements are $C(a)$ and $P(a,b)$ where a and b are constants.

Table 2. Mappings

Mappings	DL Notation	FOL translation
(a) Class inclusion	$\mathcal{P}_1:C_1 \sqsubseteq \mathcal{P}_2:C_2$	$\forall X, \mathcal{P}_1:C_1(X) \Rightarrow \mathcal{P}_2:C_2(X)$
(b) Property inclusion	$\mathcal{P}_1:P_1 \sqsubseteq \mathcal{P}_2:P_2$	$\forall X \forall Y, \mathcal{P}_1(X, Y) \Rightarrow \mathcal{P}_2(X, Y)$
(c) Domain typing of a property	$\exists \mathcal{P}_1:P \sqsubseteq \mathcal{P}_2:C$	$\forall X \forall Y, \mathcal{P}_1(X, Y) \Rightarrow \mathcal{P}_2:C(X)$
(d) Range typing of a property	$\exists \mathcal{P}_1:P^- \sqsubseteq \mathcal{P}_2:C$	$\forall X \forall Y, \mathcal{P}_1(X, Y) \Rightarrow \mathcal{P}_2:C(Y)$

Query answering is a two-step process: first, queries are rewritten in a set of more specific queries. The set of all the rewritings of a query can be obtained from the conjunctions of the rewritings of each relation (property or class) of the query. Then, every rewriting is evaluated to get corresponding data. Users can pose unary, conjunctive or disjunctive queries. In case of conjunctive queries, rewritings are obtained from the conjunctions of the rewritings of each relation of the original query. In case of disjunctive queries, each disjunction is managed as a unary query.

3 Exploiting SomeRDFS Reasoning

SomeRDFS reasoning, in particular, query answering can offer an automated support for discovering new mappings. We propose in this section a method to guide the ontology matching process based on query answering. Query answering is used to generate mapping shortcuts and to identify relations, denoted target relations, which are starting points in the mapping discovering process. These relations allow identifying relevant mapping candidates limiting that way the matching process to a restricted set of elements. Discovered mappings can be relevant or not according to the strategy involved in the PDMS. Thus, in a first sub-section we present the different strategies that can be followed by a peer. In the next sub-section, we present how mapping shortcuts and target relations can be identified using query answering. In the last sub-section, we describe the techniques used to obtain a set of relevant mapping candidates from a set of target relations and corresponding to a given strategy.

3.1 Strategies of a Peer

A PDMS can be seen as a very large data management system with a schema and data distributed through, respectively, the union of the peer ontologies and mappings, and

the union of the peer storage description. It can also be viewed as a system where many peers each with its own ontology, mappings and data, choose to share data. In any case each peer has to access knowledge of the other peers. Having more mappings can be beneficial for three reasons. It is a way to access new data sources and so obtaining richer answers. It is a way to allow more precise queries assuming users are able to pose queries using the relations in mappings belonging to the vocabulary of distant peers. Finally, it is a way to make the PDMS steadier because less dependant of the comings and leavings of the peers in the network. Thus, any peer may decide to increase the number of its mappings. It can decide to look for new mappings whatever they are (the *default* strategy denoted S_1) or to look for particular mappings: either new mappings involving peers not yet logically connected to it (the *not yet connected peers* oriented strategy denoted S_2) or mappings involving peers already logically connected to it (the *connected peers* oriented strategy denoted S_3). Two peers are logically connected if there exists a mapping between their two ontologies. The choice of one of these strategies depends on the number of already connected peers and on the number of mappings involving a given peer.

3.2 Using Query Answering

Mappings Shortcuts Discovery

A mapping shortcut is a composition of mappings. Mapping shortcuts consolidate PDMSs by creating direct links between indirectly connected peers. We could imagine automatically combining mappings in order to obtain shortcuts. Indeed, given a peer \mathcal{P} systematic queries corresponding to each of its relation allows to identify mapping shortcuts involving each of them. However, this method generates a lot of traffic on the network and all the mappings obtained this way are not always useful. Mapping shortcuts are useful when some peers disappear from the PDMS. However, they do not lead to more answers and they add caching in the rewriting process.

We propose a two-step automatic selection process. We first identify potentially useful mappings shortcuts exploiting query answering. In this step, the goal is to retain only mappings which would be useful in the rewriting process with regard to the queries really posed by users to the peer \mathcal{P} . Then we propose a second selection step based on filtering criteria.

To achieve the first step we need to distinguish the rewriting and evaluation phases of query answering. Query answering will not be a unique and global process anymore but two connected processes which can be separated if needed. Indeed, users do not always find the right needed relations in the ontology of the queried peer. In that case, they choose other relations among the relations of the queried peer. However, if a more specific relation is queried all the required data will not be obtained. On the other hand, if a more general relation is asked, all the required data will be obtained but these data will be mixed to others. For example, a user may need asking \mathcal{P}_1 for instances of *SteelSculptor*. Such a query can not be posed because of the lack of the *SteelSculptor* relation in \mathcal{P}_1 . The user could decide to ask for a more general relation, for example $\mathcal{P}_1:Artist(X)$. Rewritings obtained involve $\mathcal{P}_2:SteelSculptor(X)$ which is the relation he is interested in, but also $\mathcal{P}_2:WoodSculptor(X)$ and $\mathcal{P}_2:GlassSculptor(X)$. The evaluations of these two later

relations are not needed with respect to the user's expectations. Considering rewriting as a process different from evaluation allows the user to examine the results of the rewriting phase in order to select which rewritings have to be evaluated. The fact that the user selects rewritings that have to be evaluated is a good indicator of the relations he is really interested in. Thus, we propose to analyze the interactions between users and peers and to add mappings that are direct specialization links between the (more general) queried relation and the one the user has chosen to be evaluated. In this example, it would be $\mathcal{P}_2:SteelSculptor(X) \Rightarrow \mathcal{P}_1:Artist(X)$ added to \mathcal{P}_1 . We consider that this mapping is a useful mapping shortcut. Note that if the user asks for the evaluation of several relations several mapping shortcuts will be proposed. Furthermore, in this article we do not describe discovering of mapping shortcuts based on conjunctive queries because of space limitations.

The second selection step is based on the strategy of the peer and potentially exploits filtering criteria defined by the administrator of this peer. Indeed, according to the strategy S_2 or S_3 chosen by a peer \mathcal{P} only a subset of the mapping shortcuts will be considered. Then a peer may want to operate a finer selection using additional filtering criteria. The usable criteria are specific to each peer but are limited. They concern either the kind of user (member of a particular group or of a given category: permanent users, temporary users, users making an intensive use of the peer, ... assuming that the group and the category are given when a user registers) who posed the query which originated the mapping (user-criterion) or the kind of relation belonging to \mathcal{P} involved in the mapping (relation-criterion). The favored relations can be indicated one by one or according to their level in the hierarchy. We can, for instance, favor mappings establishing a connection with the n last levels in the class or property hierarchy of the ontology. A value is associated to each criterion, 1, 0.5 or 0, depending on whether the involved mapping has to be more or less favored. Let us note that the same mapping can be obtained several times from different queries potentially posed by different users. The weight of the user-criterion may be different from one mapping to another but the weight of the relation-criterion will always be the same. Thus, we propose a relevance measure for the mapping shortcuts which takes into account the weight of each additional filtering criterion but also the number of times that the mapping was obtained. Table 3 gives the value of the relevance measure of a mapping shortcut m_j when this mapping has been obtained n times given a sample of studied shortcut mappings composed of M elements. $W(U_{i,j})$ is the weight of the user-criterion for the occurrence i of the mapping m_j . $W(R_j)$ is the weight of the relation-criterion for the relation R_j .

Table 3. Relevance measure of the mapping m_j with n occurrences

n : # occurrences of m_j	relevance measure
$n \geq 80\% \times M$	1
$50\% \times M \leq n < 80\% \times M$	$\text{Max}\left(0.5, \frac{\sum_{i=1}^n W(U_{i,j}) + n \times W(R_j)}{2n}\right)$
$n < 50\% \times M$	$\frac{\sum_{i=1}^n W(U_{i,j}) + n \times W(R_j)}{2n}$

Identification of Target Relations

In SomeRDFS mappings are the key notion for establishing semantic connections between the peers ontologies. They are used to rewrite queries posed in terms of a local ontology and rewritings are then run over the ontology of logically connected peers. That way, distant peers may contribute to answers. However, when a user interrogates the PDMS through a peer of his choice answers may be unsatisfactory because of a lack of specialization mappings. The problem may originate from relations called target relations, which are blocking points in query answering because they are an obstacle in achieving the strategy its peer has chosen to implement. Our objective is to identify them and to consider them as starting points in the ontology matching process. As the different strategies that we consider (cf. Section 3.1) rely on the number of logically connected peers and on the number of specialization mappings the definition of a target relation will be based on a counting function. That function will differ according to the strategy of the peer and also according to the method used to count. Indeed, given a relation R of a peer \mathcal{P} , the number of distant relations involved together with R in RDF(S) statements, either specialization mappings of \mathcal{P} or locally inferred statements, can be calculated either with regard to the knowledge of the peer \mathcal{P} or with regard to rewritings obtained from queries. This is also true when given a relation R of a peer \mathcal{P} we want to compute the number of distant peers corresponding to relations involved in specialization mappings of R or locally inferred statements. The result of the counting function will be compared to a threshold that will be fixed by the administrator of the peer. When the value of the function is lower than the threshold the relation will be a target relation. We first give a general definition of a target relation. We will then precise the general definition to handle all the different cases.

Definition 1 (Target Relation) $\mathcal{P}_1:R_1$ is a target relation iff $f(\mathcal{P}_1:R_1) < t$, f being a counting function and t a threshold.

In Table 4, we precise the definition of the function f for the relation R_1 of the peer \mathcal{P}_1 according to the strategy chosen by the peer and according to the method used to count (cf Section 3.1).

Note that the target relations obtained using the counting function C_2 will be different from the target relations obtained using C_1 . Indeed C_2 takes into account distant relations which belong to rewritings produced by connected distant peers but not distant relations coming from disconnected peers and C_1 does the opposite.

If the strategy of \mathcal{P}_1 is the *default* strategy S_1 and if C_1 is used the result of $f(\mathcal{P}_1:R_1)$ is the number of distant relations specializing R_1 according to the mappings of \mathcal{P}_1 or specializing another relation R_k of \mathcal{P}_1 with $R_k \Rightarrow R_1$ locally inferred. Using C_2 the result of $f(\mathcal{P}_1:R_1)$ will be the number of distant relations belonging to the rewritings of R_1 . If this number of distant relations is lower than the threshold t then R_1 will be a target relation.

If the strategy of \mathcal{P}_1 is the *not yet connected peers* oriented strategy S_2 and if C_1 is used the result of $f(\mathcal{P}_1:R_1)$ is the number of distant peers involved in specialization mappings of R_1 or in statements specializing another relation R_k of \mathcal{P}_1 with $R_k \Rightarrow R_1$ locally inferred. If this number of distant peers is lower than the threshold t then R_1

Table 4. Definition of $f(\mathcal{P}_1:R_1)$

Method used to count Strategy of a peer	C_1 (with regard to the knowledge of \mathcal{P}_1)	C_2 (based on rewritings)
S_1 (default strategy)	$ \{\mathcal{P}_i:R_j / [\mathcal{P}_i:R_j \Rightarrow \mathcal{P}_1:R_1]$ or $[\exists \mathcal{P}_1:R_k \text{ such that } \mathcal{P}_1:R_k \Rightarrow \mathcal{P}_1:R_1 \text{ can be inferred and } \mathcal{P}_i:R_j \Rightarrow \mathcal{P}_1:R_k]\}$	$ \{\mathcal{P}_i:R_j \in RW\} $ where RW is the query rewriting set of $Q \equiv \mathcal{P}_1:R_1$
S_2 (not yet connected peers oriented strategy)	$ \{\mathcal{P}_i:R_j / \exists \mathcal{P}_i:R_j \text{ such that } [\mathcal{P}_i:R_j \Rightarrow \mathcal{P}_1:R_1]$ or $[\exists \mathcal{P}_1:R_k \text{ such that } \mathcal{P}_1:R_k \Rightarrow \mathcal{P}_1:R_1 \text{ can be inferred and } \mathcal{P}_i:R_j \Rightarrow \mathcal{P}_1:R_k]\}$	$ \{\mathcal{P}_i:R_j / \mathcal{P}_i:R_j \in RW\} $ where RW is the query rewriting set of $Q \equiv \mathcal{P}_1:R_1$
S_3 (connected peers oriented strategy)	$\min_i (\{\mathcal{P}_i:R_j / \mathcal{P}_i:R_j \Rightarrow \mathcal{P}_1:R_1$ or $[\exists \mathcal{P}_1:R_k \neq 1] \text{ such that } \mathcal{P}_1:R_k \Rightarrow \mathcal{P}_1:R_1 \text{ can be inferred and } \mathcal{P}_i:R_j \Rightarrow \mathcal{P}_1:R_k]\})$	$\min_i \{\mathcal{P}_i:R_j \in RW\} $ where RW is the query rewriting set of $Q \equiv \mathcal{P}_1:R_1$

will be a target relation. Using C_2 the result of $f(\mathcal{P}_1:R_1)$ will be the number of distant peers involved in the rewritings of R_1 .

If the strategy of \mathcal{P}_1 is the *connected peers* oriented strategy S_3 , R_1 will be a target relation if there is at least one peer involved in a low number of specialization statements of R_1 . Thus, if C_1 is used, $f(\mathcal{P}_1:R_1)$ provides the minimum number of relations of a given distant peer specializing R_1 according to the mappings of \mathcal{P}_1 or specializing another relation R_k of \mathcal{P}_1 with $R_k \Rightarrow R_1$ locally inferred. If C_2 is used, $f(\mathcal{P}_1:R_1)$ will provide the minimum number of distant relations which belong to the rewritings of R_1 and which are involved in the mappings of \mathcal{P}_1 .

3.3 Obtaining a Set of Relevant Mapping Candidates

Our objective is to use target relations in order to identify relevant mapping candidates, limiting that way the matching process to a restricted set of elements. In this section, we propose methods to discover new mappings from target relations. These methods are performed by a given peer given its target relations.

Target relations can allow discovering relevant mapping candidates according to two scenarios. In the first scenario (cf Figure 1), let us consider $\mathcal{P}_1, \mathcal{P}_2$ and \mathcal{P}_3 three peers with C_1, C_2 and C_3 three classes and the following mappings: $\mathcal{P}_1:C_1(X) \Rightarrow \mathcal{P}_2:C_2(X)$ and $\mathcal{P}_3:C_3(X) \Rightarrow \mathcal{P}_2:C_2(X)$, each known by the two involved peers.

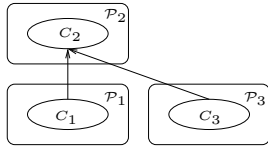


Fig. 1. Scenario 1

From the point of view of \mathcal{P}_1 $C_1(X)$ is a target relation because there is no distant relation specializing $C_1(X)$. The query $Q_4(X) \equiv \mathcal{P}_1:C_1(X)$ has no rewriting. That target relation is interesting since $\mathcal{P}_1:C_1(X) \Rightarrow \mathcal{P}_2:C_2(X)$ is a mapping in \mathcal{P}_1 , $Q_5(X) \equiv \mathcal{P}_2:C_2(X)$ could be a query posed to \mathcal{P}_2 by \mathcal{P}_1 . The obtained rewritings would be $\mathcal{P}_1:C_1(X)$ and $\mathcal{P}_3:C_3(X)$ and looking for mappings between all the relations belonging to this set of rewritings is relevant. Indeed, it could allow to discover the mapping $\mathcal{P}_3:C_3(X) \Rightarrow \mathcal{P}_1:C_1(X)$ making that way a connection

between all the relations belonging to this set of rewritings is relevant. Indeed, it could allow to discover the mapping $\mathcal{P}_3:C_3(X) \Rightarrow \mathcal{P}_1:C_1(X)$ making that way a connection

between \mathcal{P}_3 and \mathcal{P}_1 . Note that, according to this scenario 1, the peers \mathcal{P}_1 and \mathcal{P}_2 can be the same, and \mathcal{P}_2 and \mathcal{P}_3 too.

In the second scenario (cf Figure 2) let us consider \mathcal{P}_1 and \mathcal{P}_2 two peers, $\mathcal{P}_1:C_1$, $\mathcal{P}_2:C_2$ and $\mathcal{P}_2:C_3$ three classes. $\mathcal{P}_2:C_2(X) \Rightarrow \mathcal{P}_2:C_3(X)$ is a statement in \mathcal{P}_2 . $\mathcal{P}_2:C_2(X) \Rightarrow \mathcal{P}_1:C_1(X)$ is a mapping in \mathcal{P}_2 and \mathcal{P}_1 .

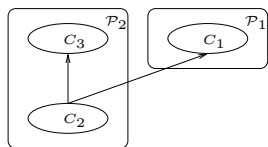


Fig. 2. Scenario 2

From the point of view of \mathcal{P}_2 $C_2(X)$ and $C_3(X)$ are target relations because there is no distant relation specializing $C_2(X)$ nor $C_3(X)$. The query $Q_6(X) \equiv \mathcal{P}_2:C_3(X)$ has only one local rewriting which is $\mathcal{P}_2:C_2(X)$. No distant relations belong to the rewritings. This scenario is also interesting since $\mathcal{P}_2:C_2(X) \Rightarrow \mathcal{P}_1:C_1(X)$ is a mapping in \mathcal{P}_2 , it could be relevant to look for mappings between

$C_1(X)$ and $C_3(X)$, two relations which subsume $C_2(X)$. It could allow to discover the mapping $\mathcal{P}_1:C_1(X) \Rightarrow \mathcal{P}_2:C_3(X)$ establishing a connection between \mathcal{P}_2 and \mathcal{P}_1 usable to rewrite $\mathcal{P}_2:C_3(X)$.

Let us note that the $\mathcal{P}_1:C_1(X) \Rightarrow \mathcal{P}_2:C_2(X)$ mapping in scenario 1 and the $\mathcal{P}_2:C_2(X) \Rightarrow \mathcal{P}_2:C_3(X)$ mapping in scenario 2 can be locally inferred in \mathcal{P}_1 and \mathcal{P}_2 , respectively. Furthermore, these two scenarios are elementary and could be combined in order to deal with more complex ones. These two scenarios use those target relations as starting points for the identification of relevant mapping candidates. However, all target relations will not allow finding relevant mapping candidates. Thus, we just consider target relations with regard to the two scenarios described above.

For each target relation we look for sets of mapping candidates, denoted MC . Our approach is based on the idea that it is relevant to look for connections between relations if they have common points. In our setting the common point that we are going to consider is a common relation, either more general or more specific. The construction of the set of mapping candidates can be achieved according to two processes.

Specific Candidates Algorithm: This algorithm is performed for target relations with one or more general relations, R_g , according to the knowledge of its peer (according to the ontology or to the mappings). This scenario is represented in Figure 1 with C_2 in the place of R_g . In that case, we propose to pose the query $Q(X) \equiv R_g(X)$ in order to obtain its rewritings. The set of the rewritings is MC . It is composed of relations that are more specific than R_g .

General Candidates Algorithm: This algorithm is performed for target relations R_s with several (at least two) more general relations according to the knowledge of its peer (according to the ontology or to the mappings). This scenario is represented in Figure 2 with C_2 in the place of R_s . In that case, all the more general relations are members of the set MC .

4 Conclusion and Perspectives

In this paper we have presented how SomeRDFS query answering can offer an automated support for discovering new mappings. In particular, we have shown that query answering in a decentralized setting can be used to select elements which are relevant to be matched when the number of elements to be matched is a priori huge and when

no peer has a global view of the ontologies in the network. Our approach is based on query answering and filtering criteria. It applies to any system with large amounts of data organized according to local RDF schemas and which provides a communication infrastructure based on query rewriting: PDMSs but also, in a more general way, other decentralized systems such as networks of existing websites or local databases.

Currently, we implemented the identification process of potentially useful mapping shortcuts according to the default strategy S_1 . We also implemented the identification process of target relations and mapping candidates according to each of the strategies and counting methods introduced in this paper. We have a running prototype, Spy-Where, providing mapping candidates in SomeRDFS peers. The first experiments show the relevance of our approach. In a near future, we plan to integrate suitable alignment techniques. In a previous work, we addressed the problem of taxonomy alignment when the structures of the taxonomies are heterogeneous and dissymmetric, one taxonomy being deep whereas the other is flat [12]. We are going to explore the suitability of these techniques to our new context in order to propose extensions or adaptations really suited to the SomeRDFS PDMSs setting. Then we plan to evaluate our approach more completely. Future research includes also considering coherence issues due to the integration of discovered mappings among older ones.

References

1. Adjiman, P., Goasdoué, F., Rousset, M.C.: SomeRDFS in the semantic web. *Journal on Data Semantics* (2006) p. 158–181
2. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. *VLDB J.* **10**(4) (2001) 334–350
3. Euzenat, J., Shvaiko, P.: *Ontology matching*. Springer-Verlag, Heidelberg (DE) (2007)
4. Kalfoglou, Y., Schorlemmer, M.: Ontology mapping: the state of the art. *The Knowledge Engineering Review* **18** (2003) p. 1–31
5. Halevy, Y., Ives, G., Suciu, D., Tatarinov, I.: Schema mediation for large-scale semantic data sharing. *The VLDB Journal* **14**(1) (2005) 68–83
6. Tatarinov, I., Ives, Z.G., Madhavan, J., Halevy, A.Y., Suciu, D., Dalvi, N.N., Dong, X., Kadiyska, Y., Miklau, G., Mork, P.: The piazza peer data management project. *SIGMOD Record* **32**(3) (2003) 47–52
7. Herschel, S., Heese, R.: Humboldt discoverer: A semantic p2p index for pdms. In: *DIS-Web'05*. (June 2005)
8. Castano, S., Ferrara, A., Montanelli, S.: H-match: an algorithm for dynamically matching ontologies in peer-based systems. In: *SWDB 2003*, Berlin, Germany (September 2003)
9. Bernstein, P.A., Green, T.J., Melnik, S., Nash, A.: Implementing mapping composition. In: *VLDB*. (2006) 55–66
10. Besana, P., Robertson, D.: How service choreography statistics reduce the ontology mapping problem. In: *ISWC/ASWC*. (2007) 44–57
11. Aberer, K., Cudré-Mauroux, P., Hauswirth, M.: The chatty web: emergent semantics through gossiping. In: *WWW*. (2003) 197–206
12. Reynaud, C., Safar, B.: When usual structural alignment techniques don't apply. In: *The ISWC'06 workshop on Ontology matching (OM-06)*. (2006)