MULTI-DATA SOURCE FUSION APPROACH IN PEER-TO-PEER SYSTEMS

Gilles Nachouki[1] and Marie-Pierre Chastang[1,2]

[1]LINA - UMR 6241, Départ. Informatique, Université de Nantes, France
Gilles.Nachouki@univ-nantes.fr

[2]Polytech'Nantes, Université de Nantes, France
Marie-pierre.Chastang@univ-nantes.fr

ABSTRACT

In this paper, a new approach for data fusion in the context of schema-based Peer-To-Peer (P2P) systems is proposed. Schema-based systems manage and provide query capabilities for (semi-)structured information: queries have to be formulated in terms of schema. Schema-based P2P are called Peer Data Management system (PDMS). A challenging problem in a schema-based peer-to-peer (P2P) system is how to locate peers who have data relevant to a given query.

Our proposal lies in application of the multi-data source fusion approach in the context of PDMS. Multi-data source schemas, distributed, shared and maintained by peers, are the basis of a semantic overlay network. The semantic overlay network and the power of Multi-data source Fusion Language (MFL) are exploited for efficient query routing towards the relevant peers. We show the design of the Peer Multi-Data source Management System (PMDMS) and we focus on the Matchmaker and routing components. We give a performance evaluation of the semantic query routing with respect to important criteria such as precision, recall, response time and number of messages. We give a performance evaluation of the semantic reconciliation between peers. We compare this result with other systems developed according to peer/super-peer approach. Finally, we show a prototype developed according to PMDMS. We build an application that shares, data between peers, in the domain of leisure such as, bank, cinema, restaurant, etc.

KEYWORDS

Peer-To-Peer, Multi-data source Fusion, Ontology, Relational database, XML technologies

1. INTRODUCTION

In the past, intensive researches related to data integration were focused on databases. A federated database (FDB) [21] is a collection of cooperating and autonomous database systems (DBs). A federated database management system (FDBMS) provides a controlled and coordinated manipulation of DBMS component. With the advent of the Web, data management

has moved away from the traditional framework to match the variety of information available on the Web. Web documents are often used for storing data over the internet, whilst XQuery language has become the standard for retrieving such documents. Nowadays, the amount of information produced in the world increases by 30% every year and this rate will only go up [14]. Peer-to-peer (P2P) systems adopt a completely decentralized approach for data sharing over the Web. An important issue in P2P systems is the choice of the approach for data placement across peers, and to ensure data availability without incurring additional overheads. Schema-based P2P systems denoted by PDMSs aim at overcoming the scalability problems of data integration systems by combining P2P and distributed database techniques. Peers join the system by providing their own schemas and matching their respective schemas to discover their acquaintances for effective data sharing.

In this paper, we consider unstructured PMDMSs (Peer Muti-Data source Management Systems) [17]. We suppose that each peer maintains a multi-data source schema describing data sources schemas (of semantically linked peers or neighboring peers) and conflicts between them. In fact, data sources present heterogeneity that consists of differences in names, types, etc. Several perceptions of the same real world lead to different schemas. To integrate the data sources together, we firstly need to solve the conflicts between their schemas. The multi-data source schema is the basis of a semantic overlay network where peers having similar elements form a semantic network neighborhood. This semantic overlay network is exploited further to address query propagation. Multi-data source Fusion Language (MFL) is provided for data sources fusion in multi-data source approach. MFL is a simple and powerful language that allows users to define the schema of the multi-data source and formulate their need in a single query. In fact, for each submitted query on the multi-data source schema, MFL will search for conflicts in the query's body. Three cases may arise: 1. If no conflict is detected, the query will be accepted and sent to only relevant peers; 2. Conflicts cannot be solved; in this case, the query will be rejected (e.g. case of homonyms conflicts); 3. Some conflicts are solved where others are not; in this case, a part of the query related to the solved conflicts will be generated and executed on relevant peers. We make the following supposition and contributions: we focus on MFL to define multi-data source schema and exchange queries between peers.

To illustrate our approach, we take the following scenario which concerns the arrival and/or departure of peers: a new peer P_a joins the PMDMS with a suggested schema S_a. P_a advertises its expertise by sending to its neighbors a Domain Advertisement $DA_a = (PID, E_a^{Xp}, T_a, \varepsilon_{acc}, TTL)$ containing the peer ID denoted *PID*, the suggested expertise E_a^{Xp}, the topic area of interest T_a and the minimum semantic similarity value (ε_{acc}). To carry out efficient communication and message forwarding among peers during domain foundation, we combine a constrained flooding algorithm that decreases duplicate queries with a *TTL* mechanism that helps reducing the radius of the discovery query coverage. When receiving a domain advertisement (e.g. DA_a), a peer P_j, interested by this domain, invokes the reconciliation process to find semantic mappings between the elements of its schema denoted $\boldsymbol{P_j^{Owl}}$ and the elements of the expertise E_a^{Xp}. The semantic links that were found are sent also to the peer P_a which can approve or reject them. If the collaboration has been accepted then each peer (e.g. P_j) search conflicts between the expertise E_a^{Xp} (being integrated by peer P_j) and the multi-data source schema maintained by peer P_j denoted $MSch_j$. These conflicts included in $MSch_j$ are used later to help the peer P_j to determine relevant peers for a query.

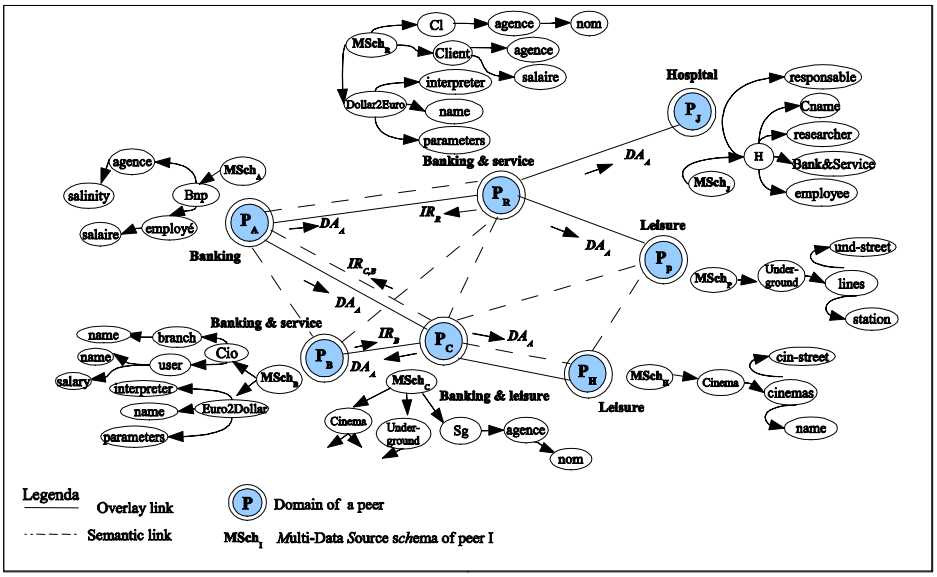

Figure1. Semantic overlay network formation in PMDMS.

In Figure 1 we show the overlay links between peers and the creation of the semantic overlay network progressively after the junction of peer A to the network. The peers P_P, P_H and P_C are linked together as they are interested in the same domain: *Leisure*. At the same time, the peer P_C is linked semantically with peers P_A and P_R as they share the same domain: banking. P_R is linked semantically with peers P_I as they share the same domain: banking services and Hospital.

The remainder of this paper is organized as follows: section 2 gives the background of this work. Section 3 introduces briefly the Multi-data source language. Section 4 shows the design of our PMDMS (Peer Multi-Data source Management Systems). Section 5 shows how to build the semantic overlay network in PMDMS. Section 6 describes our queries routing method. Section 7 shows the results of simulations concerning the routing of queries; the performance of the algorithm of semantic reconciliation between peers and compares this result with a PDMS (i.e. super-peer/peer) system. Section 8 describes briefly our prototype, a PMDMS, used in order to share data between peers, in the domain of *Leisure*. Section 9 gives related work concerning data integration. Section 10 discusses and gives some future works.

2. BACKGROUND

2.1. Basic Notions

A peer is an autonomous entity with a capacity of storage and data processing. In a computer network, a peer may act as a client or server. A P2P is a set of autonomous and self-organized Peers (P), connected together through a computer network. The purpose of a P2P network is the sharing of resources (files, databases etc.) distributed on peers by avoiding the appearance of a peer as a central server in this network. We note: $P2P = (P, U)$ where P is the set of peers and U represents links (overlay connections) between peers (e.g. P_a and P_j), $U \subseteq P \times P$. A PMDMS (Peer Multi-Data source Management System) combines P2P systems and support the Multi-data source Fusion Language (MFL) [16]. The peers in PMDMS that we consider are supposed to hold relational databases or XML documents. Let T, $T=\{T_1,...,T_P\}$ represents the set of interest themes or domains (e.g. Restaurants, Cinemas, Banking, etc.) of peers. In our case,

peers express their interest to one or several theme(s) in *T*. A peer expresses its schema in an XML document with the OWL/RDF language (e.g. $\boldsymbol{P_a^{Owl}}$, $\boldsymbol{P_j^{Owl}}$) [20]. Two peers A and J can publish in their respective schemas the same concepts or features with distinct structures and/or do not use the same vocabulary. A Peer joins the network with a *Co*ncrete data source (e.g. P_a^{Co}, P_j^{Co}). A Peer sends a Domain Advertisement (e.g. DA_a, DA_j) that contains a specification of the data schema called *expertise (e.g.* E_a^{xp}, E_j^{xp}) to share with others peers. Indeed, the schema of each peer (e.g. $\boldsymbol{P_j^{Owl}}$) is described with a set of synonyms in order to help later the reconciliation between the expertise E_a^{xp} and the schema $\boldsymbol{P_j^{Owl}}$.

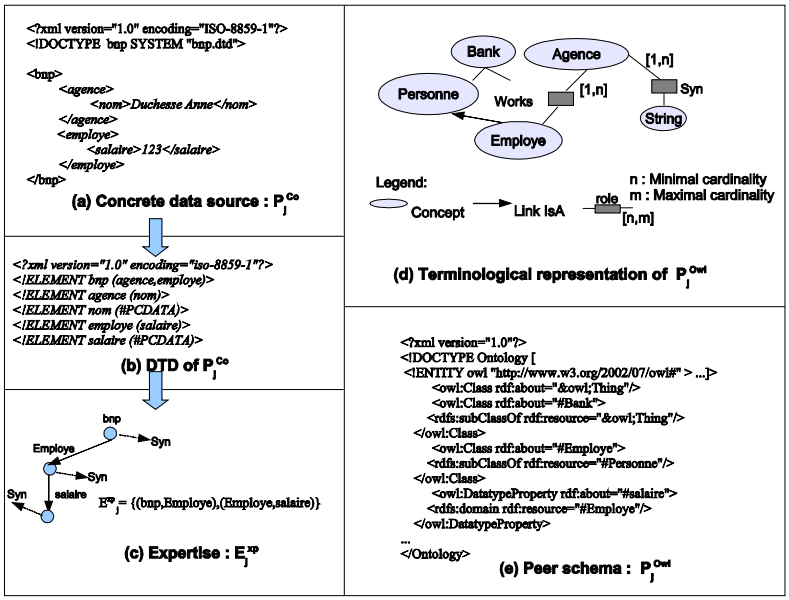

Figure 2. Peer schema, XML document, Concrete DTD and Expertise E_j^{Xp}

Figure 2(d) shows with a terminological model a part of the schema banking supported by the peer P_j. A part of this schema is expressed in figure 2(e) with OWL/RDF language (denoted $\boldsymbol{P_j^{Owl}}$). In figure 2(a), P_j has a concrete data source (P_j^{Co}), stored in an XML document, to share with other peers. The DTD in figure 2(b) is extracted from P_j^{Co}. Peer P_j publishes a part of the DTD through the expertise E_j^{Xp} given in figure 2(c). E_j^{Xp} is defined with the same structure and vocabulary used in the schema $\boldsymbol{P_j^{Owl}}$. Therefore, there is no semantic reconciliation inside the same peer between its expertise (e.g. E_j^{Xp}) and its schema (e.g. $\boldsymbol{P_j^{Owl}}$).

2.2. Expertise, Mapping and Semantic Overlay network

Whatever the data model used (e.g. relational database or XML document) to model the concrete data source P_j^{Co} provided by peer J, we extract from it the DTD. The concept of expertise was proposed initially in [11]. The expertise (E_j^{XP}) of a peer P_j is defined, in our case, as (a part of) the data source (i.e. DTD) of this peer. This expertise is expressed as a set of couples of elements linked between them by direct links. We note that: $E_j^{Xp} = \{(n_p, m_p) \in DTD \wedge r(n_p, m_p)\}$ where $r(n_p,m_p)$ means that there exists a direct link from element n_p to element m_p in the DTD. In our context, mapping is an important process: to share data between two peers P_a and P_j, it is important to begin by looking for connections between expertise of P_a (e.g. E_a^{Xp}) and the schema of peer P_j (e.g. P_j^{Owl}) or inversely between E_j^{Xp} and P_a^{Owl}. In general, the search for correspondence between two schemas S1 and S2 consists to find for each concept

in S1 (or S2) a correspondent in S2 (or S1) which is nearest semantically. We can define the concept of mapping (Map) between schemas as follows:

$$Map: S1 \Rightarrow S2, Map(c_{s1}) = c_{s2} if\ Sim(c_{s1}, c_{s2}) > \epsilon_{acc} \quad (1)$$

Where

c_{s1} : element of schema S1; c_{s2} : element of schema S2; ϵ_{acc} is the acceptable threshold; $Sim(c_{s1}, c_{s2})$ is a function, that measure the similarity between two concepts c_{s1} and c_{s2} , given as follows:

$$Sim: S1XS2 \rightarrow [0,1] \quad (2)$$

We distinguish two particular cases: $Sim(c_{s1}, c_{s2}) = 1$ describes two similar elements; $Sim(c_{s1}, c_{s2}) = 0$ describes two distinct elements. The set of Semantic Links (SL) that relates two peers P_i and P_j is denoted SL^{Pij}. The Semantic Overlay Network (SON) [6] is defined as follows:

$$SON = \cup_{i,j=1}^{|T|}(SL^{Pij}) \quad (3)$$

Where

$i \neq j \wedge SL^{Pij} = SL^{Pji}$;
$SL^{Pij} = \phi$ *means that no Semantics links between* P_i *and* P_j

SON is defined as the union of all semantic links between peers where |T| represents the total number of peers in the PMDMS network. In the next section we present briefly the Multi-data source Fusion Language (MFL).

3. THE MULTI-DATA SOURCE FUSION LANGUAGE

MFL provides two sub-languages [16]: the Multi-data source Definition Language (MDL) - used to define the multi-data source - and the Multi-data source Retrieval Language (MRL) - used to retrieve data from multi-data sources. One characteristic of MDL resides on the simplicity of the multi-data source's definition: users can give a collective name called *multi-data source name* to some data sources. In MRL, the names of data sources used in a query are mentioned. A collective name simplifies the expression of queries to some data sources. MRL extends the XQuery language in order to access multiple conflicting data sources. Furthermore, in conflicting data sources, the user's need is expressed through a single query. With MRL, it is easy to smooth out semantic data differences which often exist in autonomous data sources.

3.1. Multi-data source definition language

The purpose of MDL is to define the multi-data source schema of a peer starting from a set of data sources schemas. A multi-data source schema is a collection of data sources' schemas or multi-data sources. The following example shows the schema of a multi-data source owned by peer *Sg* denoted P_{Sg}. In this example the three peers P_{Bnp}, P_{Cio} and P_{Cl} are called remote peers.

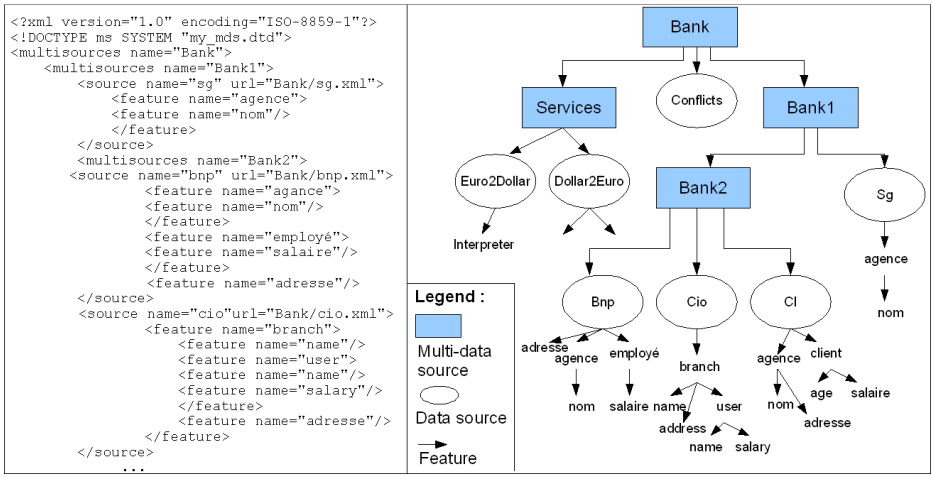

Figure 3. A part of the Banking schema $MSch_{Sg}$ owned by peer P_{Sg}

Example 1 (Multi-data source creation): consider the multi-data source given in Figure 3 describing a banking domain. In this example, the three expertises *Bnp*, *Cio* and *Cl,* represent a multi-data source called *Bank2*. The data sources *Bnp*, *Cio* and *Cl* are supposed published and shared by three remote peers denoted P_{Bnp}, P_{Cio} and P_{Cl}. A data source *Sg* is supposed published and shared by the peer P_{Sg}. The *M*ulti-data *s*ource *Sc*hema of figure 3, owned by P_{Sg}, is denoted $MSch_{Sg}$. Similarly, *Bank2* with bank *Sg* form multi-data source called *Bank1* and *Bank* is the root of the multi-data source. Conflicts between these data sources are given in a specific data source called *Conflicts* (described below). *Services* represent a multi-data source which describes two (active) data sources called *Dollar2Euro* and *Euro2Dollar* published and shared by P_{Sg}. The first service converts currencies from Dollar to Euro whereas the second one converts currencies from Euro to Dollar. In figure 3, the name of a static or active data source (e.g. *Bnp*) represents the root of a document (e.g.*Bnp*.dtd) that describes the schema of this source. In this example, the schema of data sources *Bnp*, *Cl* and *Sg* are expressed in French and the data source *Cio* in English. The structure of multi-data source *Conflicts* is detailed in the remaining of this section.

We suppose the following descriptive conflicts between elements of two data sources: synonymous, equivalence, homonymous, disjoint and scale conflicts. Two elements are considered semantically synonymous if their names are synonymous and have the same context (i.e. synonymous conflicts). Two elements are considered semantically equivalent if their names are the same and have the same context (i.e. equivalence conflicts). Such elements are connected between them by a similarity link in the Conflicts data source (given below). Two elements are linked by a dissimilarity (or difference) link if their names are the same or synonymous and have two distinct contexts (i.e. homonymous conflicts). Finally, two elements are linked by a dissimilarity link if their names are distinct and not synonymous (i.e. disjoint conflicts). A scale link describes two elements which are similar and each one uses a specific scale unit (i.e. scale conflicts). The structure of the data source Conflicts is given in Figure 4.

```
<!ELEMENT CONFLICTS (DISSIMILAR*,SIMILAR*,SCALE*,SERVICES*)>
<!ELEMENT DISSIMILAR (ELT,PATH,PATH+)>
<!ELEMENT SIMILAR (Node*)>
<!ATTLIST SIMILAR Sim CDATA #REQUIRED>
<!ATTLIST SIMILAR Conf CDATA #REQUIRED>
<!ELEMENT Node (PATH,ELT,UNIT?)>
<!ELEMENT SCALE (Node*)>
<!ATTLIST SCALE TYPE CDATA #REQUIRED>
<!ELEMENT SERVICES (SERVICE*)>
<!ATTLIST SERVICES TYPE CDATA #REQUIRED>
<!ELEMENT SERVICE (NAME,PATH,CONVERT)>
<!ELEMENT NAME (#PCDATA)>
<!ELEMENT CONVERT (UNIT1,UNIT2)>
<!ELEMENT UNIT1 (#PCDATA)>
<!ELEMENT UNIT2 (#PCDATA)>
<!ELEMENT ELT (#PCDATA)>
<!ATTLIST ELT Sim CDATA #REQUIRED>
<!ATTLIST ELT Conf CDATA #REQUIRED>
<!ELEMENT PATH (#PCDATA)>
 <!ELEMENT UNIT (#PCDATA)>
```

Figure 4. Structure of Conflicts data source

3.2. Multi-data source retrieval language

We present briefly the Multi-data source Retrieval Language (MRL) and we give an example of a query using this language. An MRL query form is defined as follows:

Use (multi-)datasource$_1$ *name*$_1$ *[,(multi-)datasource*$_j$ *name*$_j$*]**
Allow $ <semantic variables>
(E)XQuery query
Close name$_1$ *[,name*$_j$*]**

The clauses *Use*, *XQuery query* and *Close* are mandatory in MRL Queries, whereas the clause *Allow* is optional. The clause *Use* determines the scope of the query and connects to data sources for processing whilst the clause *Close* disconnects from data sources. The *name* is a given alias for either a data source or multi-data source and the clause *Allow* permits the declaration of semantic variables. Through these variables, the user declares his/her intention to access data, in a given query, that are semantically similar and differently named. The (E)XQuery query can be formulated like a query w.r.t. to XQuery language or as an EXQuery query that allows an active data source to be called.

Example 2 (One semantic variable): Select in $MSch_{sg}$ the name of the branches in the two data sources *Bnp* and *Cio*.

Q: *use bnp b,cio c*
allow $x=nom,name
for $a in document("MSch*$_{sg}$")/bank/bank1/bank2, $b in $a/*/$x*
return <result>$b/text()</result>
close b,c

This query is a semantic query that uses only one semantic variable (denoted x) in its body. The following section shows the design of the Peer Multi-Data source Management Systems (PMDMS) based on MFL.

4. PMDMS ARCHITECTURE

In this section, we describe the architecture of our PMDMS [17]. Figure 5 depicts the main components of our PMDMS. As shown, it is composed of five main components: MDSManager, Peer Interface (PI), Query Routing (QR), MatchMaKer (MMK) and Peer Communications Services (PCS). MDSManager (Multi-data source Management Systems) is composed principally of three components: Wrapper, Mediator and Interfaces. More details about the design of MDSManager are given in [16].

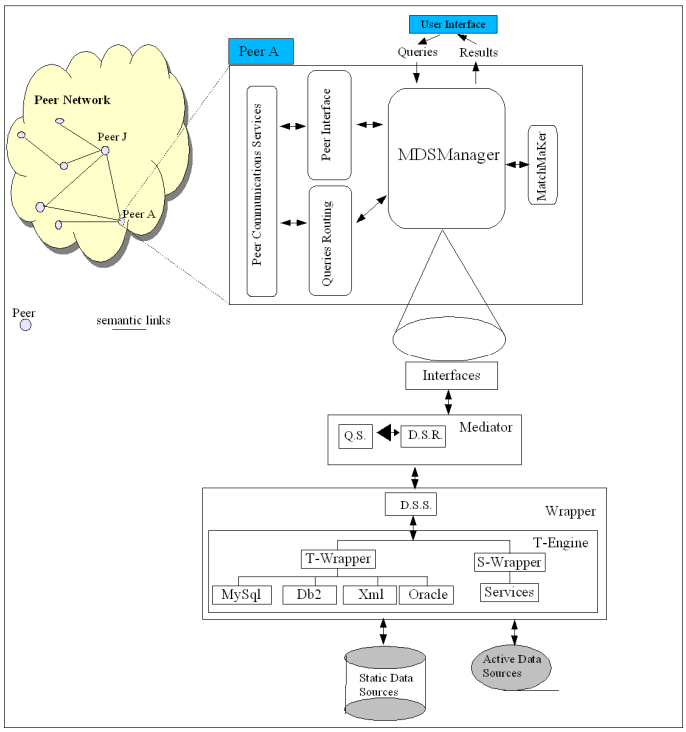

Figure 5. Design of PMDMS system

The Peer Interface (PI) receives expertise or queries from remote peers. When a peer P_j receives an expertise from peer P_a, the interface submits this expertise to the MatchMaker MMK in order to match the received expertise E_a^{Xp} of the peer P_a with its schema P_j^{Owl}. If the two peers accept to collaborate then each one (e.g. P_j) searches conflicts between the expertise of the other peer (e.g. E_a^{Xp}) and its *M*ulti-data *S*ource *sc*hema (e.g. $MSch_j$). Finally, each peer (e.g. P_j) stores the expertise of the other peer (E_a^{Xp}) and the conflicts found in its multi-data source schema $MSch_j$. When PI receives an MRL query from a peer, the interface submits this query to the MDSManager component for processing over its local data sources.

The Query Routing (QR) component of a peer (e.g. P_j) is invoked when the user submits an MRL query and that the scope of this query (i.e. the clause *Use*) refers to data sources owned by remote peers (e.g. P_a). For such query, MDSManager of peer P_j search for conflicts in the query's body in order to generate pertinent MRL query. To check a query is pertinent the following treatment is allowed: if no semantic conflict is detected inside the query then the query is considered pertinent; if there are semantic conflicts and none of them can be resolved, then the query will be rejected (e.g. case of homonyms conflicts) without sending it to remote

peers; if there are some resolvable conflicts, then the part of the query related to these conflicts will be generated and executed; MDSManager returns a pertinent query to QR component. Then, this component sends the pertinent query to only relevant peers.

The Peer Communications Services (PCS) allows the peers to communicate to each other. PCS component is achieved by JXTA. JXTA defines a common set of protocols for building P2P applications. JXTA offers developers the means to design any kind of overlay network that suits to the needs of their applications.

The following section shows our approach in order to discover (semi-)automatically the conflicts existing between expertise in a multi-data source schema (e.g. $MSch_{sg}$) owned by a peer (e.g. P_{sg} in figure 3). Our approach can be summarized in two steps: 1. the first step [18] consists to search similarities between the expertise of a remote peer (received by P_j) and the schema $\boldsymbol{P_j^{Owl}}$ of peer P_j. This step can be (semi-)automatic because it sometimes (e.g. when ambiguities arise) user is asked to validate some similarities between elements; 2: the second step consists to generate the Conflicts data source. The data source Conflicts is used later (in the next section) in order to help a peer (e.g. P_j) to select relevant peers for a pertinent query. This step is entirely automated and it's based on XML and XSLT technologies.

5. SEMANTIC RECONCILIATION BASED ON ONTOLOGY

Ontology is an explicit specification of a conceptualization [9]. In this section we start by introducing briefly the benefit of using ontologies in the domain of data sources integration. Then, we present briefly the principle of reconciliation between peers in PMDMS.

5.1. Ontology and the integration field

In the early stage of the Web, information was shared as HTML pages. These pages were designated to be read only by a human user. The first language designed by the consortium W3C in the domain of Web Semantic is the RDF (Resource Description framework) language [20]. RDF is an XML language used for describing metadata and for facilitating their treatment by specific applications programs. RDFS (RDF Schema) language was developed after in order to give RDF more expressive power. However, many limitations restrict the ability to express knowledge. Indeed, it is not possible to carry out an automated reasoning on knowledge modeled using RDFS. To overcome this lack, a new language for Web called OWL was developed (Ontology Web Language) [15]. OWL is based on logic description. Using OWL, one can describe the knowledge about a domain in terms of a set of classes and a set of properties. Classes represent entities of interest in a specific domain and a property represents a feature (i.e. data type property) of an entity or a relationship between entities (i.e. object properties). OWL, like RDF, is based on XML language. OWL provides tools for comparing and reasoning on classes, their features and the relations between them. It gives a great ability to interpret the web content because it contains a wide range of vocabulary and a full semantic formal. The W3C provided three types of languages to better express OWL: Lite, DL (Description Logic) and Full.

Ontologies are used in integration tasks to describe the semantics of data sources. In [22], three different approaches are discussed to use ontologies in the process of integration of data sources: single ontology, multiple ontologies and hybrid ontologies. In the single ontology approach all data sources are related to global ontology. In the second approach, each data source has its own ontology. In this case mapping between ontologies are necessary for

integrating data sources. The hybrid approach is a combination of the two previous ones. In [7], the authors describe the role of ontologies in data integration in two settings: central and P2P. The single approach is appropriate for *GaV* (Global As View) systems and hybrid approach is more appropriate for *LaV* (Local As View) [12] systems; hybrid peer-to-peer system, where a global ontology exists in a super-peer can also use hybrid ontology approach.

5.2. Principe of reconciliation between peers in PMDMS

In PMDMS, each peer P_j has a concrete data source P_j^{Co} (to share with others peers) and a peer schema ($\boldsymbol{P_j^{Owl}}$). P_j extracts the DTD from P_j^{Co} and defines its expertise E_j^{Xp}. Every element e_i in E_j^{Xp} is affiliate to an element in $\boldsymbol{P_j^{Owl}}$ which has a set of keywords *syn* (e_i).

To discover semantic equivalences, between expertises of peers, the Matchmaker of a peer (e.g. P_j) is designed with three main components (figure 6): 1. The first component is the *Expertise Enrichment*: it extracts elements from an expertise (e.g. E_a^{Xp}) and matches them with the elements of schema of peer P_j (e.g. $\boldsymbol{P_j^{Owl}}$) [18]. The result of this step is denoted EE_a^{Xp} *(Enrichment Expertise of peer A)*. EE_a^{Xp} associates each element in E_a^{Xp} to an element in $\boldsymbol{P_j^{Owl}}$ which is the most nearest semantically; 2. The second component is the Wrapper: it takes in input the Enrichment Expertise (EE_a^{Xp}) and transforms it into an instance of the ontology $\boldsymbol{P_j^{Owl}}$ expressed with the language OWL/RDF. This transformation is based on a defined template (e.g. Ontology.xslt) not shown in this paper; 3. The third component is the mediator. His main roles are: ***a.*** to create each instance returned by the wrapper in the ontology $\boldsymbol{P_j^{Owl}}$; ***b.*** to regroup together, for each Data Type Property (DTP) of a class all its instances in $\boldsymbol{P_j^{Owl}}$. This step may be accomplished by querying directly the ontology with one of the languages of ontologies (i.e. DLQuery etc.) or using XML and XQuery language. This result is stored in an XML document named *GroupInst.xml*. ***c.*** to deduce similarities (dissimilarities) by transforming the document *GroupInst.xml*, using a template *Similarity.xslt* (*Dissimilarity.xslt*) not shown in this paper. The result returned by Mediator is stored in the schema $MSch_j$ of peer P_j in a specific data source called *Conflicts*.

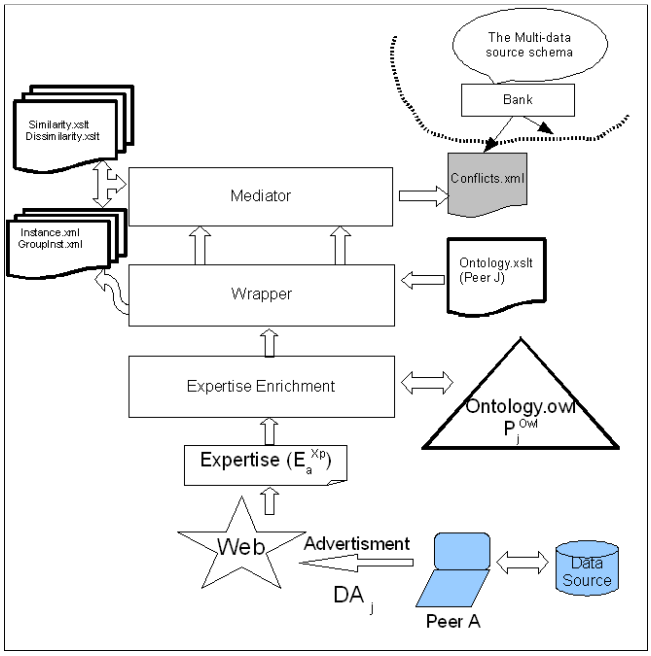

Figure 6. Reconciliation between Peers

A part of this Conflicts data source is given in figure 7. This data source *Conflicts.xml* is used in the following section (query routing) to generate pertinent MRL queries and to route queries toward relevant peers.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE CONFLICTS SYSTEM "conflicts.dtd">
<CONFLITS>
<DISSIMILAR>
    <ELT>name</ELT>
   <PATH>Bank/Bank1/Bank2/cio/branch/user</PATH>
   <PATH>Bank/Bank1/Bank2/cio/branch</PATH>
</DISSIMILAR>
<SIMILAR>
  <Node>
   <PATH>Bank/Bank1/Bank2/cio/branch</PATH>
   <ELT>name</ELT>
  </Node>
  <Node>
   <PATH>Bank/Bank1/Bank2/bnp/agence</PATH>
   <ELT>nom</ELT>
  </Node>

</SIMILAR>
...
</SERVICES>
</CONFLICTS>
```

Figure 7. A part of the Conflicts data source

6. QUERY ROUTING

6.1. Principle

In this section, we show the steps of routing pertinent queries in PMDMS, based on MFL and semantic overlay network, to only relevant peers. Firstly, we check that, each submitted query is pertinent or semantically coherent. If a query is not pertinent then we search in the scope of the query parts which are coherent semantically and in some cases the query is refused. Secondly, pertinent query is represented under the form of a tree. Using this tree we extract the subject of the query. The subject of a query is an abstraction of the query in term of elements that it contains. We compare the subject of the query with the expertise of peers in the clause *Use* in order to deduce the set of relevant peers. Then, sub-queries are generated and sent to relevant peers for processing and results are returned to the user.

6.2. Generation of pertinent queries

A query submitted by user through its peer is firstly checked by the system and converted into a *pertinent* query. Pertinent query means that the query should be coherent semantically as a whole or not be semantically in conflicts within its own properties. Consequently, the parts of the query which are not consistent are ignored thereafter. We illustrate our propos with the following example.

Example 7 (Pertinent query generation): Let us now consider the query (*Q*) expressed over the schema $MSch_{Sg}$ of figure 3. This query researches the agencies of banks *Cio* and *Sg* located in *Londre* street.

Q:<MRL>
use cio c,sg s
allow $x.y.z=branch.name.address, agence.nom.adresse
<XQuery>
for $a in document("MSch$_{sg}$")/bank/bank1,
$b in $a/*/$x[$z='Londre']
return
<result>$b/$y/text()</result>
</XQuery>
close c,s
</MRL>

Here, the clause *Allow* declares three of semantic variables x, y and z where x, y and z designate respectively the three couples of values (*branch*, *'agence'*), (*name*, *'nom'*) and (address, 'adresse'). Moreover, the designator name is a multiple identifier since it designates the name of a branch and the name of a user in the data source *Cio*. Similarity and dissimilarity conflicts are detected through query processing in the Conflicts data source. An example of conflict resolution is shown here: the meaning of *'nom'* of an *'agence'* in *Sg* is found to be different from the name of a user in *Cio* but it is similar to the name of a branch. In this case, we keep only the possibility for the variable 'y' to take the two features: *'nom'* of an *'agence'* and *name* of a *branch*. The meaning of '*adresse*' and *address* are found similar in the Conflicts data source. To obtain a pertinent query, the domain of the variable y is restricted to only the feature *name* of a branch and the *'nom'* of an *'agence'*. Two equivalent pertinent queries (elementary queries) Pq_1 and Pq_2 are generated:

Pq_1: <MRL>
use cio c
<XQuery>
for $a in document("*MSch$_{sg}$*")/bank/bank1,
$b in $a/*/branch[address='Londre']
return <*result*>$b/name/text()</*result*>
</XQuery>
close c
</MRL>

Pq_2: *<MRL>*
use sg s
<XQuery>
for $a in document("*MSch$_{sg}$*")/bank/bank1,
$b in $a/*/agence[adresse='Londre']
return <*result*>$b/nom/text()</*result*>
</XQuery>
close s
</MRL>

The first MRL query corresponds to the substitution of *branch*, *name and address* respectively to the three semantic variables x, y and z. In the same perspective, *'agence', 'nom'* and 'adresse' are related respectively to x, y and z in the second query.

6.3. Processing of pertinent queries

We distinguish two main steps in the processing of a pertinent query which are: subject extraction and relevant peers' selection.

Step 1. **Subject extraction.** A pertinent MRL query is firstly expressed as a tree. This tree is composed of: 1. a sub-tree which represents the scope of the query (clause Use); 2. a sub-tree (optional) which represents the semantic variables (clause Allow); and 3. a sub-tree FLWR that is an expression of the XQuery clauses: For, Let (optional), Where (optional) and Return. Consider the pertinent query Pq_1 given in Figure 8(a). The tree of this query is illustrated in Figure 8(b).

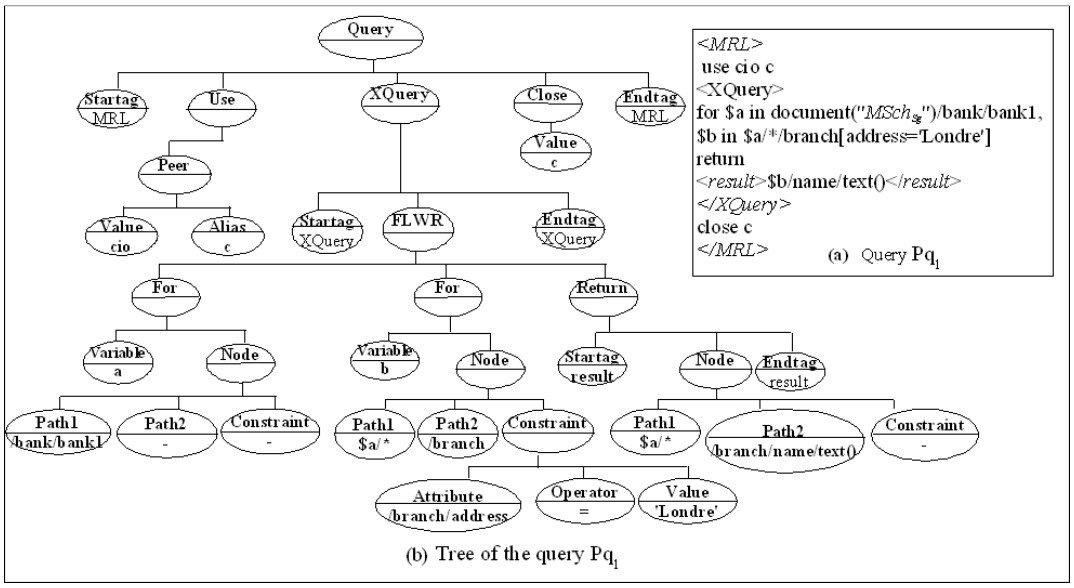

Figure 8. Representation of query with tree

A peer extracts the subject of a query starting from its tree. The subject of a query represents an abstraction of this query in terms of elements belonging to its tree. It's defined as follows:

$$Sub^{Q}_{MSch_P} = \{X(A_q, C_q) \in Tree_Q | (Query/*/X/node/path2/A_q, Query/* /X/node/constraint/attribute/C_q)\} \quad (8)$$

Where *Sub* designates the Subject of Q. *Node* represents two paths and a constraint: the first path ($Path_1$) is defined over the multi-data source schema held by peer P (denoted $MSch_P$); the second path ($Path_2$) represents a path specified inside a specific data source (e.g. *Cio*) in $MSch_P$. A constraint is represented under the form: (Attribute Operator Value) or (Attribute Operator Attribute). X takes one of the following values: For (F), Let (L), Where (W) or Return (R). For Pq_1 the subject of this query is given as follows:

$$Sub^{Pq1}_{MSch_{sg}} = \{F(/branch, /branch/address), R(/branch/name/text(), -)\}.$$

Step 2. **Relevant peers selection.** For a pertinent query Q for which we extract its subject, the system measures the capacity of a peer (e.g. P_a) -among the set of peers specified in the clause

Use of the query- to process it. The capacity of P_a to process a query is done by matching the subject of the query $Sub^{Q}_{MSch_j}$ to the expertise of peer P_a (E_a^{Xp}). The selection of relevant peers is based on the following function Cap_{Sim}: this function measures the similarity between E_a^{xp} and $Sub^{Q}_{MSch_j}$. Cap_{Sim} is based on the *Similarity* function that research in the Conflicts data source of $MSch_j$ the similarities between an element *e* in E_a^{xp} and an element *s* in $Sub^{Q}_{MSch_j}$.

$$Cap_{Sim} = \frac{1}{|Sub^{Q}_{MSch_j}|}\left(\sum_{s\in Sub^{Q}_{MSch_j}} \; \sum_{e\in E_a^{xp}}^{Similarity}(s,e)\right) \quad (9)$$

Where

$$Similarity(s,e) = \begin{cases} 1, if \;\; s\ is\ found\ similar\ to\ e \\ 0, if \;\; s\ is\ found\ dissimilar\ to\ e \end{cases}$$

If Cap_{Sim} is equal to one then the Peer P_a is relevant for the query Q. Inversely, if Cap_{Sim} is equal to zero the similarities between elements in $Sub^{Q}_{MSch_j}$ and E_a^{xp} are all distinct in the Conflicts data source. Between these two cases cited above, we can found that some elements in E_a^{xp} which are similar with elements in $Sub^{Q}_{MSch_j}$whilst others elements are different. In this case, a peer is considered relevant if $Cap_{sim} > \mu_{acc}$. The value of the acceptable-threshold determines the peers for which the pertinent query will be sent. This value is determined by the user according to the level of precision he/she wants to obtain.

Example 8 (Relevant Peers selection): Select the names and addresses of agencies *in data sources Bnp (P_{bnp}), Cl (P_{cl}) and Sg (P_{sg})* given in $MSch_{sg}$ (figure 3):

```
Q:<MRL>
use bnp b,cl c,sg s
<XQuery>
for $a in document("MSch_Sg ")/bank/bank1,
$b in $a/*/agence
return
<result>
<nom>$b/nom/text()</nom>,
<adresse>$b/adresse/text()</adresse>
</result>
</XQuery>
 close b,c,s
</MRL>
```

This query is checked pertinent in the scope of peers P_{bnp}, P_{cl} *and* P_{sg}. The subject of this query inferred from the tree of Q is given as follows:

$$Sub^{Q}_{MSch_{Sg}} = \{F(/agence,-), R(/agence/nom/text(),-), R(/agence/adresse/text(),-)\}$$

We measure the capacity of each peers (i.e. *Bnp*, *Cl* and *Sg*) to process Q. We find that the capacity of the peer *Cl* is equal to 1. On the other side, the capacity of the peer *Sg* (*Bnp*) is roughly equal to 0.66. It depends on the value of μ_{acc} -introduced by the user- the peer *Sg* (*Bnp*) could be considered relevant or not.

6.4. Sub-trees generation

This step generates a sub-tree for each relevant peer found among the set of peers specified in the clause *Use* of the query. Indeed, a sub-tree is a tree sent by the Query Routing component to a relevant peer for processing. A sub-tree is defined like the tree's format given above. A sub-tree does not contain the specific path ($Part_1$) which is depends on the multi-data source schema of the peer where the query is submitted. A sub-tree is translated by the remote peer (i.e. the peer that receives the sub-tree) into an MRL query. For its treatment, this query is supported directly by the MDSManager component of the remote peer.

7. PERFORMANCE EVALUATION

In this section, we show the results, of using semantic network and MFL in the context of peer-to-peer systems. These results are obtained with a SimJava-based simulator. For our evaluation we used recall and precision metrics borrowed from Information Retrieval (as given below), as well as the number of messages per query trace and response time. We apply them at the peer selection level with number of peers ranging from 100 to 3000.

$$Recall = \frac{|Reponse_{pertinent} \cap Reponse_{found}|}{|Reponse_{pertinent}|} \tag{10}$$

$$Precision = \frac{|Reponse_{pertinent} \cap Reponse_{found}|}{|Reponse_{found}|} \tag{11}$$

Looking at the figure 10, our simulation shows a slight increase in response time with the increasing number of peers. This phenomenon can be explained by the fact that in our approach, a peer makes the necessary calculations to the choice of targets before sending its query.

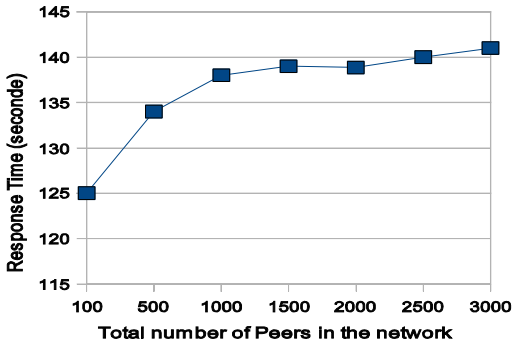

Figure 10. Response Time

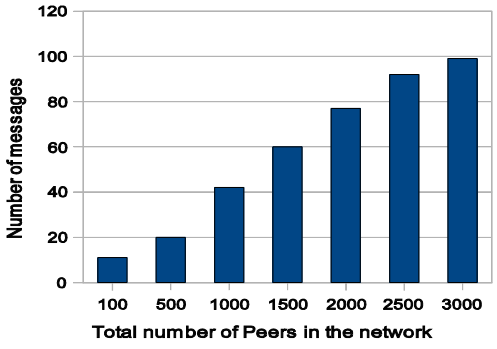

Figure 11. Number of messages

Figure 11 shows the average number of messages exchanged to answer a query. In our experiment, each peer randomly chooses data sources. The number of messages increases with the total number of peers in the network because the number of peers by topic is higher. Our architecture allows the direct shipment of query to the target peer without any intermediary.

The recall in Figure 12 is very low in our simulation. The recall lowers up to 1000 peers and then stabilizes. The reason for the low recall comes from the fact that a peer is linked directly to peers selected in the user's query. The selected peers have expertise with a great affinity. So, a

request is only sent to neighbours who have been selected for this application. With our architecture, a peer has a limited number of neighbours and it cannot, therefore, send a request to any peer in the network. This phenomenon explains a low result for our architecture which tends to decline slightly.

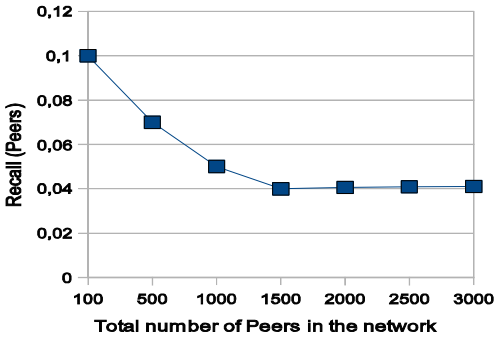

Figure 12. Recall Rate

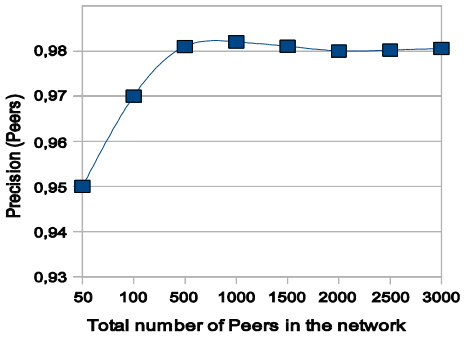

Figure 13. Precision Rate

In contrast to recall measure, in Figure 13, the precision is very high, which means that the query return very little irrelevant data. Roughly, all data returned are therefore relevant. If we analyze our results, we find that the accuracy is very high, at around 0.98. The reason for the very high accuracy is the use of MRL for describing multi-data source schema and for expressing queries. Indeed, an MRL query is converted into pertinent queries and sent to only relevant peers.

Figure 14 shows an evaluation of the algorithm used by the Matchmaker for expertise enrichment: it returns the total time used to compare the elements of two trees (i.e. DTD). We consider two settings:

- ***Setting A:*** the algorithm calculates the similarity of elements without taking into account the similarity already found between elements of their contexts. This is our baseline.
- ***Setting B:*** the algorithm calculates the similarity of two elements based on the similarity already found.

Setting A shows that the time necessary to compare two trees increases quickly when the number of elements in the trees increases (until 7000 elements). Setting A is stopped, when the number of nodes becomes roughly 4000, because the time increases dramatically. Setting B shows that the time is still acceptable and increase constantly.

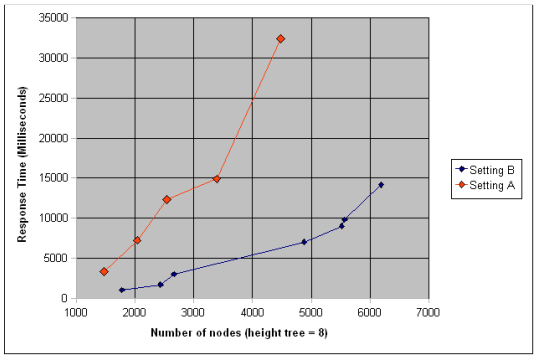

Figure 14. Response Time

8. APPLICATION

We present briefly a prototype of a PMDMS based on Jxta. This prototype is now experimented at a local network. In this section we have chosen, a P2P application, to share data between users in the domain of *Leisure*. In this application we consider only two peers denoted $peer_1$ and $peer_2$. The first peer $peer_1$ has two data sources denoted *Company* (*'Entreprise'* in French) and *Restaurant*; the second peer $peer_2$ has a data source *Cinema*. Each one of the two peers wishes to share its data with the other peer. Figure 15 shows the interface of our prototype PMDMS system. In this figure, we find that $peer_2$ has at right side top a multi-data source *Leisure* composed of three data sources *Cinema, Company* and *Restaurant*. *Company* and *Restaurant* are published and shared by $peer_1$. Figure 15 shows a query over the multi-data source schema $MSch_{Peer2}$ in order to find the names of all companies, restaurants and cinemas. This query is checked pertinent. $Peer_2$ generates: 1. a sub-tree; this sub-tree is translated into an elementary mono-data source query (MRL query) that concerns the data source Cinema. This query is executed locally by $peer_2$. 2. A sub-tree; this tree concerns the remote peer $peer_1$. It asks $peer_1$ to select the names of Companies and Restaurants from its local data sources. This tree is converted by $peer_1$ into MRL query over $MSch_{Peer1}$.

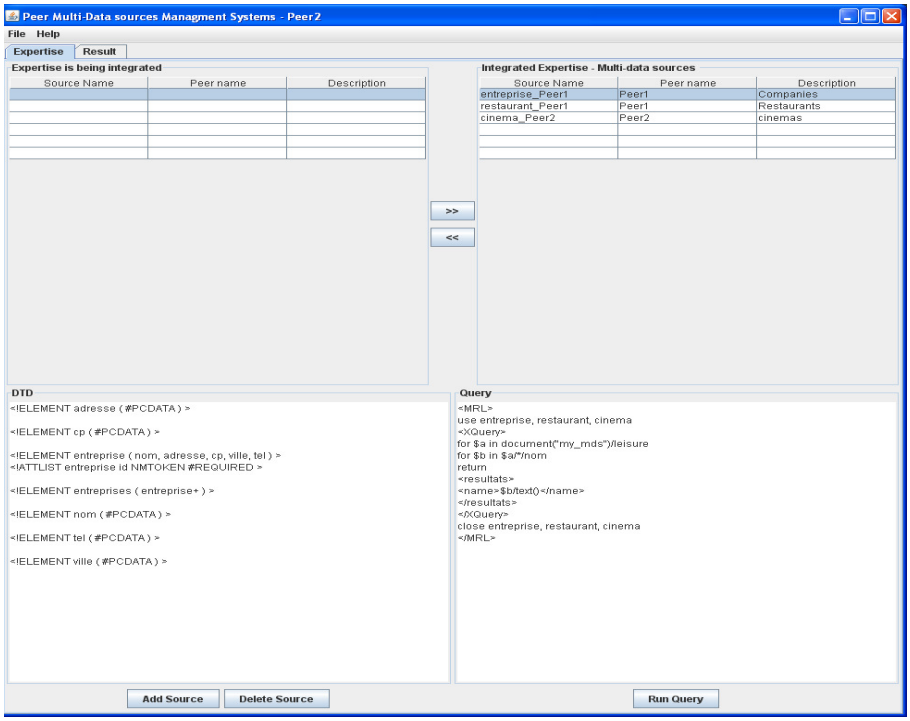

Figure 15. MRL query in P2P context

9. RELATED WORKS

Many researches in data integration's domain based on mediators use the mapping method between mediated schema and integrated data sources schemas. Two main approaches are Global As View (GAV) and Local As View (LAV) [12]. Many works have been done in order to integrate data sources in P2P context where the definition of a unique mediated schema is unrealistic because of the autonomy and volatility of peers. Data sources present heterogeneity that consists of differences in names, data structures, types, scale etc. To integrate all these data

sources together, we need first to solve their conflicts. Many works have studied the problem of conflicts in the domain of integration of data sources, we quote [4][13][19].

In P2P context, AXML (Active XML) [2] is proposed to integrate web services in an XML document. Piazza [10] integrates sources that are semi-structured data based on XML data model and XQuery language. Peers, interested by exchanging data, establish semantic links between them. SenPeer [8] is a P2P system where data sources are expressed using various data models (e.g. Relational, XML etc.). SenPeer assumes a super-peer network where peers are connected to super-peers according to their semantic domains expressed with ontology. Super-peers exchange messages to discover semantic links between their domains. In [5] authors propose indices tables for queries routing. In [3] authors propose a system that supports community formation by aggregating peers with similar interests. APPA [1] is a P2P system that makes the assumption that peers wishing to cooperate, e.g. for the duration of an experiment, agree on a Common Schema Description (CSD). Given a CSD, a peer schema can be specified using views. This is similar to the LAV approach in data integration systems, except that, in APPA, queries at a peer are expressed in terms of the local views, not the CSD. Another difference between this approach and LAV is that the CSD is not a global schema, i.e. it is common to a limited set of peers with common interest.

The main contribution of PMDMS's routing mechanism is based on the use of MFL and the use of independent schemas/ontology rather than a single shared schema / ontology. Furthermore, we use techniques combining semantic descriptions of peers in a multi-data source schema with dynamic schema matching to build a semantic overlay, on top of the underlying physical network.

10. CONCLUSION

The information society needs an efficient access to the available information which is often heterogeneous. In order to make information sharing efficient, some technical solutions have been proposed. The concept of distributed database has been introduced in order to organize a collection of multiple logically bound databases spread across a computer network. The Peer-to-Peer infrastructure is an emergent paradigm offering new opportunities for the conception of large scale distributed systems. The approach, described in this paper, integrates heterogeneous and conflicting data sources distributed on PMDMS. In this paper, we showed how to reconciliate between heterogeneous data sources. We gave a performance evaluation of the semantic query routing with respect to important criteria such as precision, recall, response time and number of messages. We gave a performance evaluation of the semantic reconciliation between peers. Our approach, compared to others system based on peer/super-peer design such as SenPeer system, presents an advantage in the number of messages exchanged over the network and the average time needed to obtain responses. The recall in our approach is less important and the precision is better. This is due to MRL as, unlike others approaches, our approach is driven initially by the choice by users of a set of data sources to integrate in its query. We showed a prototype developed using Java (JXTA) implementation and according to PMDMS. In future work, we plan to enhance more the performance of queries.

REFERENCES

[1] R. Akbarinia and V. Martins. Data management in the APPA P2P system. Journal of Grid Computing, 5(3), 303-317, 2007.

[2] O. Benjelloun, Active XML: A Data-Centric Perspective on Web Services. PhD dissertation, Paris XI University, Orsay, France, 2004.

[3] S. Castano, A. Ferrara, S. Montanelli, G. Varese, Semantic Coordination Collective Intelligence, in: Proceeding of the MEDES 2009

[4] S. Castano, A. Ferrara, S. Montanelli, C. Quix, H-MATCH: an Algorithm for Dynamically Matching Ontologies in Peer-based Systems, in: Proceedings of the VLDB International Workshop on Semantic Web and Databases (SWDB), 2003, pp. 231-250

[5] A. Crespo, H. Garcia-Molina, Routing indices for peer-to-peer systems, in: Agents and Peer-to-Peer Computing, 2004, pp. 1-13

[6] A. Crespo, H. Garcia-Molina, Semantic Overlay Network for peer-to-peer systems, in: Proceeding of ICDSC, 2002, pp.49

[7] F. Cruz, X. Huiyong, The Role of Ontologies in Data Integration, in: Journal of Engineering Intelligent Systems, Volume 13, 2005, pp. 245-252

[8] D.C. Faye, G. Nachouki, P. Valduriez, Semantic Query Routing in SenPeer, a P2P Data Management System, in: NBIS, 2007, pp. 365-374

[9] T. R. Gruber, A translation approach to portable ontologies, in: Knowledge Acquisition, 5(2), 1993, pp. 199-220

[10] A.Y. Halvey, Z.G. Ives, P. Mork, I. Tartarinov, Piazza: Data Management Infrastructure for Semantic Web Applications, in: Proceedings of the twelfth international conference on World Wide Web, 2003, pp. 556- 567

[11] P. Hasse, R. Siebed, F. Van Harmelen, Expertise-Based peer selection in peer-to-peer networks. Knowl. Inf. Syst., 15(1), 2008, pp. 75-107

[12] A.Y. Halevy, Answering queries using views: A survey, in: VLDB Journal, 10(4), 2001, pp. 270-294

[13] V. Kashyap, A. Sheth, Semantic and Schematic Similarities between Database objects: A Context-based Approach, in: Journal of the VLDB, 5(4), 1996, pp. 276-304

[14] X. Luna Dong, F. Naumann, Data Fusion – Resolving Data Conflicts for Integration, in: proceeding of the VLDB, 2009

[15] D.L. McGuinness, F. Van Harmelen, OWL Web Ontology Language Overview, in: W3C Recommendation, W3C, 2004

[16] G. Nachouki, M. Quafafou: Multi-data source fusion in : Journal of Information Fusion (Elsevier) 9(4), 2008, pp. 523-537

[17] G. Nachouki: Multi-Data Source Fusion in PDMS, in : VLDB-DBISP2P, 2008, pp. 68-80

[18] G. Nachouki, M. Nachouki, M.P. Chastang: Semantic reconciliation in peer multi-data source management system, in: IDEAS 2009, pp. . 326-329

[19] E. Rahm, P.A. Bernstein, A Survey of Approaches to Automatic Schema Matching, in: Journal of the VLDB 10(4), 2001, pp. 334-350

[20] RDF: Resource Description Framework http://www.w3.org/RDF/

[21] A.P. Sheth, J.A. Larson, Federated database systems for managing distributed, heterogeneous, and autonomous databases, in: ACMComputing Surveys (CSUR), 22(3), 1990, pp. 183-236

[22] H. Wache, T. Vögele, U. Visser, H. Stucken schmidt, G. Schuster, H. Neumann, and S. Hübner, Ontology-Based Integration of Information - A Survey of Existing Approaches, in: Proceedings of the IJCAI-01 Workshop on Ontologies and Information Sharing, 2001

Authors

Gilles Nachouki is assistant professor at Nantes University in France. He received a PhD in Computer Science from the University of Toulouse. His interest domain concerns distributed databases. Currently his principal works lie in the domain of data management in peer-to- peer systems.

Marie-Pierre Chastang is assistant professor at Nantes University. She received a PhD in Computer Science from the University of Rennes (France). Her interest domain concerns data warehouses and distributed databases.