Thematic Schemas Building for Mediation-based Peer-to-Peer Architecture [1]

Nicolas Lumineau [2] Anne Doucet [2] Stéphane Gançarski [2]

Laboratory of Computer Science of Paris VI (LIP6)
Paris, France

Abstract

During last years, mediation tools and peer-to-peer systems have allowed an important evolution for data sharing. Mediators are now mature techniques to share structured and heterogeneous data distributed through a reasonable number of nodes. Peer-to-peer architectures open new ways to build very large and dynamic networks allowing to share unsctructered data as files indexed by some keywords. We propose here to exploit the complementarity of these approaches to efficiently share structured and heterogeneous data distributed through a large set of nodes. We propose an unstructured peer-to-peer architecture handling interactions between a large set of mediators and simplifying the process of schema exchanges. We focus on the dynamic building of mediation schemas which are personalized for user needs in order to query the network. To validate our approach, we have implemented a prototype, MenT2, which integrates several schemas via mediator interactions in a simulated network.

Key words: Peer-to-Peer, Mediator interoperability, Mediation schema building

1 Introduction

Scientific applications in computer science need to manipulate huge amounts of heterogeneous data, distributed on a large number of remote sites. Exploiting those resources requires an homogeneous access to the different sources and structured queries to retrieve data corresponding to different criteria.

Mediation tools such as [13,6,10,15] are a solution which scale up to a few ten sources. The principle of mediation is to integrate schemas published

[1] This research is done in the context of the PADOUE project (`http://www-poleia.lip6.fr/padoue`) financed by ACI GRID (`http://www-sop.inria.fr/aci/grid/public`)

[2] Email: `FirstName.LastName@lip6.fr`

This is a preliminary version. The final version will be published in Electronic Notes in Theoretical Computer Science
URL: `www.elsevier.nl/locate/entcs`

by data sources into a global schema, available for applications. Structured queries over this global schema are rewritten in terms of local schemas using wrappers, then splitted into sub queries over local schemas which are sent to the relevant data sources. Results are then transferred to the mediator which integrates them before sending the final result to the application. Elaborating such a global schema is the main scientific lock for mediator scalability. Indeed, to build a global schema, all sources must be known, and the localization of a data requires querying all the mediators. The bottleneck generated for each query handling is the main limitation of such approaches.

Peer-to-peer systems (P2P) are nowadays very popular, mainly due to the growing interest for file-sharing application on the Internet, such as Napster, KaZaa or Edonkey. The main principle of Peer-to-peer is not only that each node in the network can be used as a data server and as a client, but also that nodes are dynamically organized according to nodes connections or disconnections. Because of this dynamicity, building a global schema is not possible, and each node has only a partial knowledge of the network, its neighborhood. Messages are propagated from neighbors to neighbors until relevant information is found. Various organizations for P2P systems are proposed: pure P2P, based on flooding such as Gnutella [8], hierarchical such as *Super-Peers* [24], or structured by Distributed Hash Tables (DHT) such as P-Grid [1] , Chord [22] or CAN [18]. However, these systems are insufficient for scientific applications since they only provide data sharing at file level, and a poor query language, usually based on file name search only.

Our proposal is to combine peer-to-peer architecture to guarantee scalability and mediation tool to ensure a transparent data access. Unlike existing proposals [16,2] which assume that users know all the concepts available in the network, and which dynamically maintain mappings between local and remote schemas, we propose to build mediation schemas. The originality of our proposal is to semantically enrich schemas with meta-information like thematic, temporal or localization information in order to ease schema exchanges and to provide user with a personalized schema. Our strategy fits with geographical and environmental applications, whose needs are to develop multidisciplinary data sharing, e.g. hydrologists and climatologists with town planner about flooding risks, geologists and physicists with oceanographers or petroleum companies. For these multidisciplinary applications, schema sharing is essential in order to enable users discovering new concepts.

To build mediation schema modeling data distributed through a large scale and dynamic network, we propose a two-phase mediation process: a static phase followed by a dynamic phase. The static phase allows to publish data according to thematic domains. It imposes that data providers write mediator wrappers. The dynamic phase is initiated by users. It consists in collecting and integrating schemas available in the network which correspond to users topics of interest. Thus, our system provides users with a personalized schema allowing to build retrieval queries.

Experiments are done with our prototype MEnT2 (Mediation in Two Times). This prototype runs with the relational mediator LeSelect [21] and it validates our model through simulation.

This paper is organized as follows. Section 2 gives a global overview of existing peer-to-peer architecture for data sharing. Section 3 presents our mediation-based peer-to-peer architectures. It defines the notions of published schema and thematic schema. Section 4 details the construction of thematic schemas. Our implementations are described in Section 5. Section 6 concludes and gives some perspectives about data querying.

2 Peer-to-Peer Data Sharing

Since several years, many scientific projects promote Peer Data Management Systems [3], which integrate database management and peer-to-peer systems, to study how peer-to-peer systems can be combined database management. In this context, one of the main issues is raised by the knowledge of schemas. For structured P2P architecture, [7,9] propose a solution for data sharing based on DHT. Several solutions are proposed for unstructured peer-to-peer systems. [14,5] use mediation tools for data access management, while [23,16,2,4] propose a pure semantic based solution which maintains dynamic mappings between remote nodes.

2.1 Structured Peer-to-Peer Networks

Several propositions allow structured data sharing using DHT. PIER [7] proposes an architecture for relational query processing with an index based on CAN [18]. They propose a solution to handle joins, groupings and aggregations. PinS project [9] is dedicated to metadata sharing and is based on DHT to index attribute/value couples logically distributed with Pastry [20]. Since we consider applications where data placing strategy is not possible because of sources autonomy (i.e. providers must keep their own data management and control), we do not consider structured peer-to-peer architecture, and we focus on unstructured approaches.

2.2 Unstructured Peer-to-Peer Networks

For unstructured peer-to-peer networks, we distinguish systems using mediators and systems dynamically handling mappings.

Mediator-based approaches. Edutella [14] architecture is based on RDF to describe schemas and proposes efficient techniques for RDF query evaluation through a Super-Peer architecture. The global schema is replaced by a mapping network between local schemas that allows building new mappings by transitivity. In Xyleme [5], which is dedicated to XML data, abstract DTDs are built to interface a set of DTDs dealing with common topics. The

mappings between DTDs and abstract DTDs are automatically generated by searching syntatic or semantic similarities.

Semantic-based approaches PeerDB [16] proposes a solution based on agents to dynamically handle mappings built with semantic information of schemas (set of key words). Information Retrieval techniques are used to compare relations and attributes according to these keywords in order to propagate queries towards nodes with sufficiently close schemas. The gossiping [2] gives also a solution based on dynamic mappings between local schemas expressed by queries. The neighborhood of each node is composed of nodes containing the same schema or containing schemas with known mappings. A query is rewritten according to the mappings of the remote neighbor on which the query is propagated. They define a metric for semantic comparison of queries to avoid to too many successive rewritings. Piazza [23] treats mappings between schemas to query heterogeneous sources. Each node can export data or define a "peer schema"(i.e. its own view of the network). They define mappings between two or several "peer-schemas"according to a mixed approach: Global As View and Local As View. Hyperion [4] proposes an extension of mappings in order to consider mappings between data. Triggers allow dynamically maintaining these mappings up to date.

As [5] and [23], we propose to build mediation schemas but we exploits the idea of [16] about using dictionary in order to handle two complementary sources of mappings: static wrappers of mediators and dynamic mappings dictionary. Thus we define a mediation layer adapted to dynamic network and allowing the efficient management of queries.

3 Peer-to-peer architecture based on mediation

In this section, we present some assumptions and concepts related with our application context. Then, we detail our architecture based on two mediation phases used to build a mediation schema allowing to query the network through interactive mediators.

3.1 Our context

Assumptions. To tackle the problem of data sharing in a large scale, we apply a "divide and conquer"strategy to propose a process of data sharing based on semantic labeling of schemas. Our approach implies two main assumptions about data. First, the data we want to share through the network are classifiable by a theme representing a specific domain. The set of themes are explicitly defined and shared by data providers and users. This assumption allows building a semantic vision of the network. Second, we suppose that publication standards exist for each theme. They allow defining attributes as

homogeneously as possible. Indeed, information sources are supposed to be autonomous, and no coordination between the providers should be required. The existence of publication standards is realistic especially in a context of metadata publication, which is the case for environmental or geographical metadata publication (e.g. ISO, FGDC, OpenGis,...). Note that a publication standard is not a global schema, and it does not allow a complete data integration. Based on these two assumptions, we ensure that data providers have the necessary knowledge to define the syntax (through standards) and the semantic (through themes) of their schemas. In the following, we suppose that the list of themes and standards can be consulted by all data providers and users.

Concepts. Using standards leads to consider two categories of attributes: *normalized attributes*, specified in the standard, and *specific attributes*, whose definition is free for each data provider. Note that only specific attributes can potentially create conflicts for data integration. Thus we associate a *semantic description* with attributes, through keywords expressing the concepts associated with the attribute. To avoid building a global schema, we propose to define different mediation schemas, named *thematic schemas*, *i.e.* related to a theme. Thematic schemas provide users or user communities with an access to data relevant to their topics of interest. To ease the building of these thematic schemas, we define an intermediate mediation schema, named *published schema*, containing meta-information on data, structure and data sources. Published schemas are defined for a given theme and for a given node. Moreover, they give a partially homogeneous structure of data according to publication standards.

```
<schemaQuery>                                    (1)
  <theme value="hydrology"/>
</schemaQuery>
```

A node must be able to treat two kinds of query: schema query and data query.

A *schema* query allows discovering schemas available in the network. It is represented by a XML stream as in (1), which specifies the themes interesting for the current user, here "hydrology".

A *data query* is a SQL query treated by mediators that we assume to be relational mediators. This assumption is realistic because a large proportion of existing mediators are relational. Moreover, our application framework which is done by the PADOUE project [17], is based on a relational mediator, named LeSelect [21] .

3.2 Two Phases Mediation Process

The global architecture of a node shown on Figure 1 illustrates the two phases allowing to build a thematic schema that users will use to query the network. The first mediation phase of is statically handled by the providers, while the

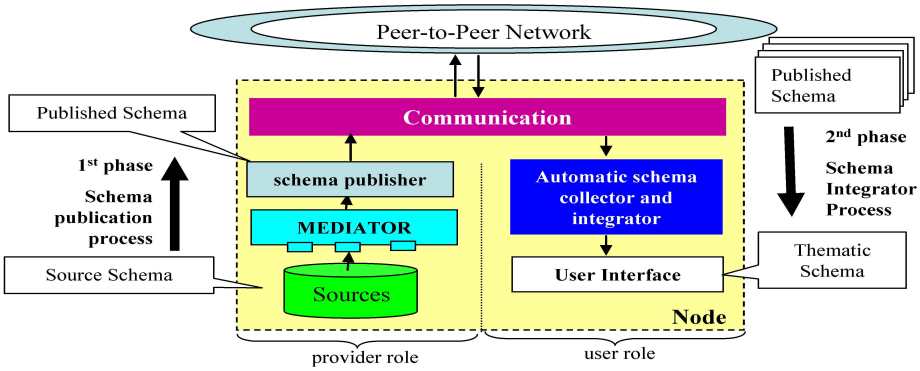

Fig. 1. The two mediation phases and associated schemas

second phase is initiated by users and dynamically handled by the system.

1) Static generation of published schemas: To ease exchanging schemas with the rest of the network, data providers generate published schemas through the schema publisher. Published schema generation for one theme consists of configuring the mediator by writing wrappers (structure publication), defining views according to the theme (semantic publication) and encapsulating the structure of the views associated with the theme, together with meta-information, in an XML stream. To realize this process, data providers know the publication standards on which the wrapper is based, and the theme catalogs allowing to define views. This phase can be viewed as a "coarse grain"mediation that allows to homogeneously define all normalized attributes.
2) Dynamic generation of thematic schemas: To generate a thematic schema, the system collects all the published schemas corresponding to the theme and currently available in the network. After being collected, these schemas are integrated. This dynamic phase provides users with a mediation schema modelling relevant and available data.

To illustrate our proposal, we consider two data providers, companyA and companyB, which decide to publish their data about dyke management, and a user on node companyC, who is interested by those data, for flood prevention. As shown in Figure 2, data provider of companyA normalizes its source schema *Doc_Dykes* to build a published schema *Dyke* associated with theme *hydrology*. This published schema is composed of two normalized attributes *langCd* and *CountryCd* and one specific attribute *lineage_stat*. The data provider of companyB normalizes his source schema *Dykes_dc* to build another published schema for hydrology composed of the same normalized attributes and the specific attribute *ftName*. This first phase of mediation is done by data providers of companyA and companyB only once, when entering to the network. Next, when the user of companyC needs all the data of relation Dyke associated with theme hydrology, the system generates on node

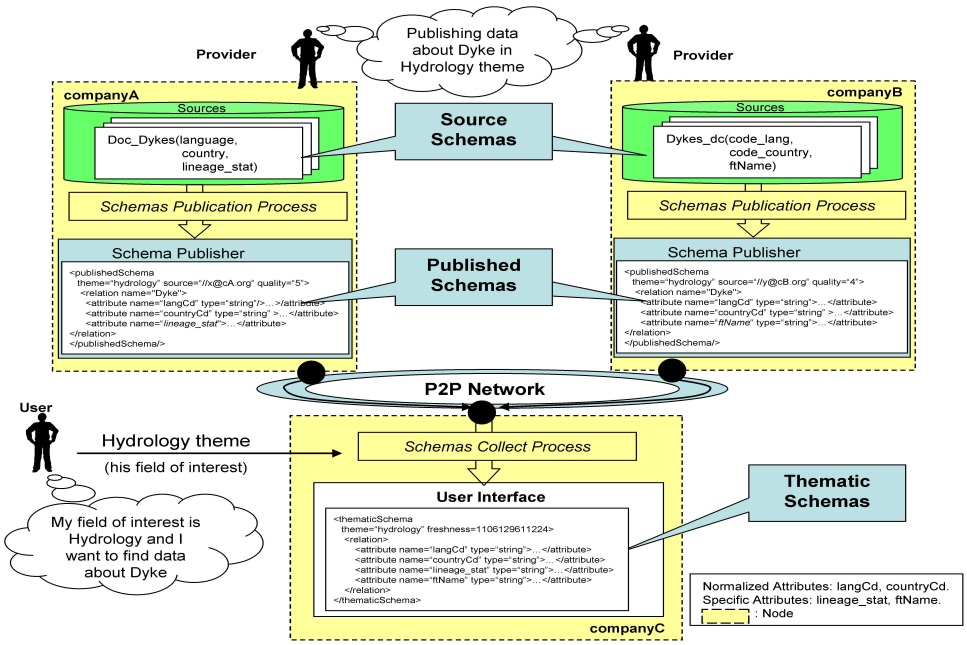

Fig. 2. Example with three nodes (two providers and one user)

companyC the thematic schema about hydrology containing the integrated schema of all published schemas. In the following, we detail the building and the management of published schemas and thematic schemas.

3.3 Thematic Schema

A thematic schema models data concerning a theme and currently available in the network. The building of this schema is initiated by the user and is dynamically done by the system. As shown in Figure 3.a., a thematic schema is characterized by a theme, here hydrology, and a freshness associated with the schema. This freshness represents the age of a thematic schema. It avoids generating a new thematic schema when it already exists on the node with a sufficient freshness. Freshness notion and thematic schema building are detailed in Section 4.

3.4 Published Schema

A published schema is built autonomously by each data provider. The purpose of such a schema is to normalize source schemas stored on the node according to a publication standard. A published schema is defined for a theme and a node. Thus, for a given theme, there are as many different published schemas as nodes storing data about this theme. As shown in Figure 3.b., a published schema on a node is characterized by a theme, here hydrology, by the node IP address, and by a quality criterion, which quantifies the number of times the current published schema has been broadcasted in the network, here five times. Indeed, published schemas are broadcasted to generate thematic schemas,

(a)

```
<thematicSchema theme="hydrology"
                freshness=1106129611224 >
 <relation name="Dyke">
  <attribute name="langCd" type="string">
   <description info="document language code"/>
   <source uri="/x@cA.org/"/>
   <source uri="//y@cB.org/"/>
  </attribute>
  <attribute name="countryCd" type="string">
   <description info="document country code"/>
   <source uri="/x@cA.org/"/>
   <source uri="//y@cB.org/"/>
  </attribute>
  <attribute name="lineage_stat" type="string">
   <description info="statement of lineage"/>
   <source uri="/x@cA.org/"/>
   <source uri="//y@cB.org/">
    <mapping as="lin_statement"/>
   </source>
   </attribute>
   <attribute name="ftName" type="string">
    <description info="format name"/>
    <source uri="//y@companyB.org/"/>
   </attribute>
  </relation>
   ...
</thematicSchema>
```

(b)

```
<publishedSchema theme="hydrology"
                 source="//x@cyA.org/"
                 quality=5>
 <relation name="Dyke">
  <normalized>
   <attribute name="langCd" type="string">
    <description info="Document language code"/>
   </attribute>
   <attribute name="countryCd" type="string">
    <description info="document country code"/>
   </attribute>
   </normalized>
   <specific>
    <attribute name="lineage_stat" type="string">
     <description info="statement of lineage"/>
    </attribute>
   </specific>
 </relation>
 ...
</publishedSchema>
```

Fig. 3. a) Example of a thematic schema built for users b) Example of a published schema broadcasted through the network

and the quality of a published schema allows to efficiently control thematic schema generation. To generate a published schema, data provider must 1) write wrappers to specify mappings between the structure of source schemas and the structure of published schemas 2) define views according to themes, and 3) specify semantic descriptions of concepts with keywords in order to allow remote users to understand the meaning of data. Finally, a published schema is generated for each theme found on the considered node. A data provider knows two information sources to build wrappers and views: the publication standard concerning the theme on which data are published, and the current thematic schema modelling data actually available on the network. The publication standards are used to specify normalized attributes and the current thematic schema is used to define specific attributes as homogeneous as possible. Note that this process of source schema normalization leads to build schemas which are not completely homogeneous because publication standards are not global schemas.

It is important to note that in a peer-to-peer context where nodes are volatile, a homogeneous definition of specific attributes can not be ensured. Indeed, data providers may define their published schema simultaneously with some others, or when disconnected from other nodes providing data about the same theme. For example, suppose that the data provider of companyA builds a published schema on hydrology. Now suppose that node companyA disconnects, and that then, the data provider of companyB defines his own published schema about hydrology. For that, he uses a thematic schema which does not

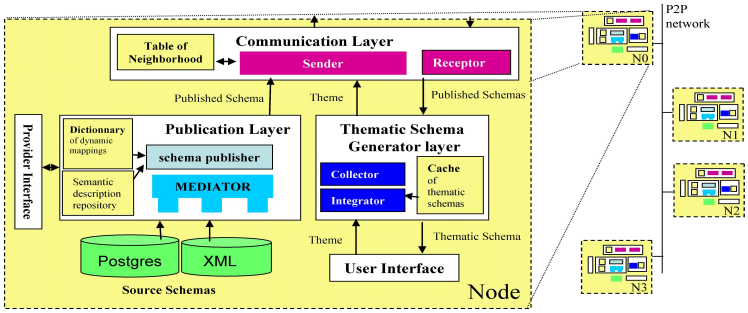

Fig. 4. Node architecture

take the published schema of companyA into account, because companyA is disconnected. Thus, companyA and companyB may give a different attribute name for the same concept. Thus, published schemas defined respectively by companyA and companyB are potentially conflicting. To detect conflicts between published schemas, we use a quality criterion for each published schema. It is a numerical value incremented each time the published schema is used to generate a thematic schema. Thus, as the node companyA is disconnected during the building process of published schema for companyB, the quality of its schema is not incremented. The difference of quality releases the analysis of schemas in order to automatically resolve the conflict. Thus, the necessary mappings between attributes are generated to build the thematic schemas giving a homogeneous view of data.

3.5 Node Architecture

Our system relies on a peer-to-peer architecture, i.e. such that each node in the network can be both data provider and user. A node can represent a unique user or a set of users in the same company. In the first case, the network topology is a classic unstructured topology comparable to Gnutella. In the second case, the topology is organized into a hierarchy, where each node is a super-peer (without metadata index), and each machine of the company is a peer. In the following, we suppose that each node is a super-peer and represents a set of users. We focus on the interoperability and the role of nodes in the peer-to-peer network.

The node architecture is illustrated in Figure 4. It contains five main layers: the publication layer, the communication layer, the thematic schema generation layer, the provider interface and the user interface.

- **The publication layer** handles the building of published schemas to allow their exchanges with remote nodes. This layer is mainly composed of a mediator and a schema publisher. The wrappers and the views configured in the mediator establish the structure and the semantic of data visible by the network. A repository contains semantic descriptions of concepts related to attributes, in order to generate published schemas as depicted in the XML stream of Figure 3.b. Moreover, a dictionary stores dynamic mappings which are not defined in wrappers and which are found during the

thematic schema generation when some conflicts between specific attributes are detected and treated. Thus, the mappings defined in the dictionary are written in published schemas in order to be considered in the future thematic schema generation.

- **The provider interface** enables wrapper and view generation for data provider. It handles all interactions between the data provider and the system.
- **The communication layer** is based on a sender and a receiver of messages (queries). Messages are treated through peer-to-peer propagation. A sent message is propagated towards the neighborhood using a neighborhood table, and a received message is treated locally and is forwarded to the neighbors.
- **The thematic schema generation layer** is detailed in Section 4. It allows integrating published schemas previously collected in the network. All thematic schemas built on a node are stored in its cache to be reused and shared between users associated with the same node. The cache is essential to manage efficiently thematic schemas. Indeed, a new thematic schema will be generated only if the cache does not store a thematic schema for the same theme and with a sufficient freshness.
- **The user interface** has two functionalities. It is used to specify the interesting theme(s) for a given user and his level of expert valuation for each theme. Moreover, it allows visualizing easily thematic schemas received in XML stream, as depicted in Fig 4a, in order to simplify the data query building.

4 Thematic schema building

As already mentioned, thematic schemas model data concerning a theme and currently available in the network. We detail in this section the generation of thematic schemas requiring to collect and to integrate all available and relevant published schemas.

4.1 Published schemas collecting

Collecting published schemas allows discovering the structure of data which are actually available in the network. This process is initiated by a user who wants to query the network. To this purpose, the user sends a schema query. The communication layer broadcasts this schema query through the network. The query is handled by each node which returns a published schema associated with the current theme, if it exists in its publication layer. Nodes which do not store relevant published schema for this theme, only propagate the query towards their neighbors. Finally, the node where the query was initiated receives a set of published schemas.

To provide users with only one mediation schema by theme, published schemas previously collected must be integrated. As already mentioned, normalized attributes do not raise specific problem in schema integration, because the publication standard ensures the homogeneity of those attributes. Thus, if published schemas have only normalized attributes, their integration is simply a strict union of their attributes. We must be more careful with specific attributes. Different published schemas may contain specific attributes with different names to define the same concept. The system must be able to define relevant mappings between these specific attributes in order to homogeneously define them in the thematic schema. Thus, we detect and resolve conflicts between specific attributes in order to merge them in the thematic schema.
Conflict detection is based on the quality of published schemas. If the quality of a published schema is lower than the others, this schema is considered as obsolete and the system chooses a published schema having the highest quality as a reference schema. Each semantic description of specific attributes in the obsolete schema is compared to the semantic description of specific attributes in the reference schema. A mapping is defined between two specific attributes if their respective semantic descriptions are close. The metric we use is based on the proportion of common words found in the semantic descriptions. When relevant mappings are found, the thematic schema is built by the schema merger. For each attribute, the thematic schema specifies its name and its type, its semantic description, the address on which it is accessible and mapping previously built, as shown in Figure 3.a. Next, the schema merger specifies the theme and the freshness of the schema (i.e. the current date) and sends it to user interface and to the cache of thematic schemas. Finally, the mappings previously found are sent to the mapping manager to update the dictionary and the quality of the published schema of concerned nodes.

Figure 5 depicts the integration of two published schemas for the theme hydrology. Published schema S1 comes from node companyA with the address //x@cA.org and S2 comes from node companyB with the address *//y@cB.org*. We suppose here that node companyB was disconnected when node companyA defined its published schemas S1, thus S1 and S2 have a different quality. This difference of quality is detected and yields the comparison between attributes of S1 and S2. The system detects that attribute *lineage_stat* of S1 and attribute *lin_statement* of S2 define the same concept. The conflict resolving algorithm chooses *lineage_stat* as attribute name in the thematic schema, since it comes from S1 which has the higher quality. Integrating S1 and S2 is depicted on the right side of Figure 5. The mapping specifying that attribute *lineage_stat* is defined as attribute *lin_statement* on node companyB is memorized in the thematic schema. This mapping is sent to the mapping manager which updates the dictionary of node companyB and its published

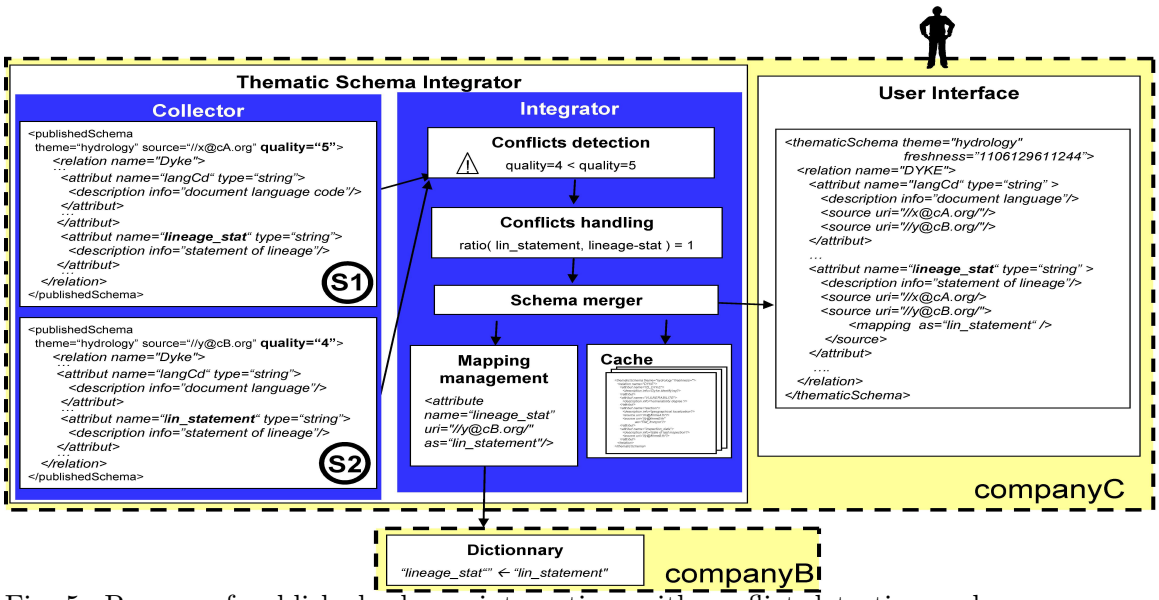

Fig. 5. Process of published schema integration, with conflict detection and management

schema about hydrology. Then, the quality of S2 becomes 5 (as the quality of S1) and the dictionary of companyB stores *lineage_stat* ← *lin_statement*. Thus, no conflict management will be necessary to build the next thematic schema by the integration of S1 and S2.

4.3 Theme popularity

As thematic schemas store information about nodes on which attributes are defined, it is important to consider the case when a theme is defined on too many nodes. In this case, the thematic schema must store information about too many nodes which is not scalable. In fact, it is comparable to maintaining global information of the network on each node. To solve this problem, we define a threshold specifying the maximum number of nodes that we can memorize for an attribute. For attributes with a number of sources greater than this threshold, we do not memorize sources and queries are propagated by flooding. This threshold ensures the scalability of our approach because no global knowledge of the network is built. Nevertheless, the consequences are important for the data query management. As some attributes may have node information and other attributes may not have, a hybrid query handling is necessary. When the clause WHERE of a SQL query, contains attribute(s) whose sources are memorized, the mediator has all information to query straightforwardly remote mediators specified in the thematic schema. Otherwise, if sources of attributes are not known, the SQL query is encapsulated in a XML stream, and is propagated through the network from neighbor to neighbor, and is handled locally by each node. Although the value of this threshold depends on mediator capacities, we claim that this threshold must evolve according to mediator load. Indeed, if the mediator only handles queries coming

from the peer-to-peer communication layer, that means the threshold is too low and it must be incremented. Thus, queries straightforwardly coming from remote mediator may appear. On the contrary, if the mediator only handles queries straightforwardly coming from remote mediators and if the time of query handling is high, that means the threshold is too high, and it must be decremented. Thus, more queries may be treated by the peer-to-peer communication layer in order to reduce the number of remote access via the mediator. The purpose of this threshold management is to dynamically maintain a query processing which adapts to load and availability of the mediator.

5 Implementations

Prototype MEnT2 (MEdiation in Two Times) has been implemented to share a set of structured, heterogeneous and distributed data via the interoperability of mediators in a peer-to-peer architecture. All implementations have been done in java. To validate the scalability of our system, we have developed a peer-to-peer simulator for unstructured networks. It allows distributing a set of logical nodes on a grid. For each logical node, we create an instance of the mediator LeSelect [21], a publication layer, a communication layer and a thematic schema generation layer. Moreover, we defined user communities with a topic of interest defined amongst a set of 8 themes. Experiments were done for 45 logical nodes distributed on a grid of 15 PC with different CPU and main memory capacities. The logical network is defined with a master node in charge of distributing logical nodes on the grid. All logical nodes are autonomous and contain provider and user agents in order to simulate human providers and users. Thus, after receiving sources schemas, provider agents automatically configure the mediator to define published schemas. Next, each user agent builds a schema query, in order to generate thematic schemas.

6 Conclusion and Future works

We propose a peer-to-peer architecture based on mediators to share structured data in a large scale network. Our motivations are based on the complementarities between peer-to-peer architecture and mediators. As global schema generation is not viable in a large scale, we propose to dynamically build thematic schemas according to user profile. These thematic schemas contain meta-information on relevant nodes which are able to handle queries. They are built according to a mediation process in two phases. The static phase allows data providers to configure their mediator to publish schemas according to a theme and to simplify schemas exchanges through the network. The dynamic phase consists in collecting and integrating exchanged schemas defined for the same theme in order to build the thematic schema modeling the data really available in the network. We validate our approach by simulation.
Our future works concern query management to extract data and logical or-

ganization of the peer-to-peer network. For data query management, we will implement the hybrid management we present in paragraph 4.3. which adapts to available meta-information in thematic schemas. Some queries will be handled directly from mediator to mediator, other queries will be handled via peer-to-peer communication layer. As our solutions are based on an important interaction between nodes, we propose a protocol of network clustering [11,12] in order to logically gather (in terms of logical neighborhood) nodes which store data concerning the same themes. We will thus improve the management of interactions between nodes of a peer-to-peer network.

References

[1] Aberer, K., Hauswirth, M., Punceva, M. *Improving Data Access in P2P systems.* IEEE Internet Computing, 6(1), 2002.

[2] Aberer, K., Cudré-Mauroux, P., Hauswirth, M. *A Framework for Semantic Gossiping.* ACM SIGMOD Record, 31(4), 2002.

[3] Aberer, K. *Special Topic Section on Peer to Peer Data Management.* ACM SIGMOD Record, 32(3), 2003.

[4] Arenas, M., et al. *The Hyperion Project: From Data Integration to Data Coordination.* ACM SIGMOD Record, 32(3), 2003.

[5] Cluet, S., Veltri, P., Vodislav. D. *Views in a Large Scale XML Repository.* In proc. of the 27th international Conference on Very Large Data Bases (VLDB 01), Roma, Italy, 2001

[6] Goasdoé, F., Lattès, V., Rousset, M-C. *The use of CARIN language and algorithms for information integration: the PICSEL system.* In the International Journal on Cooperative Information Systems, 2000.

[7] Huebsch, R., Hellerstein, J.M., Lanham, N., Loo, B.T., Shenker, S., Stoica, I. *Querying the Internet with PIER.* In Proc. of the 29th international Conference Very Large Data Bases (VLDB 03), Berlin, Germany, 2003.

[8] Jovanovic, M.A., and al. *Scalability Issues in Large Peer-to-Peer Networks - A Case Study of Gnutella.* Research report, Univ. Cincinnati, 2001.

[9] Labbé, C., Roncancio, C., Villamil, M-P. *PinS: Peer to Peer Interrogation and Indexing System.* In proc.of the 8th International Database Engineering and Applications Symposium (IDEAS 04), Coimbra, Portugal, 2004

[10] Levy, A., Florescu, D., Kossmann, D. *Querying Heterogenous Information Sources Using Source Descriptions.* In Proc. of VLDB, pp.251-262, Mumbai, India, Sept.96.

[11] Lumineau, N., Doucet, A., Defude, B. *Semantic Organization of Peer-to-Peer Network.* In proc of Semantics for Grid Databases (ICSNW 04) , Paris, June 2004. (Poster)

[12] Lumineau, N., Doucet, A., Defude, B., *VENISE: Content-based Clustering for Data Sharing in Peer-to-Peer Architecture*. In Proc. Base de Données Avancées (BDA 04), Montpellier, October 2004. (Demo)

[13] Manolescu,I., Florescu,D. , Kossmann, D. *Answering XML Queries over Heterogenous Data Sources*. In Proc. VLDB, Rome, Italie, Sept.2001.

[14] Nejdl, W., Siberski, W., Sintek, M., *Design Issues and Challenges for RDF and Schema-Based Peer-to-Peer Systems*. ACM SIGMOD Record, 32(3), 2003.

[15] Papakonstantinou, Y., Garcia-Molina, H., Widom, J. *Object Exchange Across Heterogenous Information Sources*. In Proc of ICDE, PP251-260, Taipei, Taiwan, Mars 1995.

[16] Ooi, B., Shu, Y., Tan, K.L. *Relational Data Sharing in Peer-Based Data Management Systems*. ACM SIGMOD Record, 32(3), 2003.

[17] PADOUE project : `http://www.polie.lip6.fr/ padoue`

[18] Ratsanamy S. et al., *A Scalable Content Addressable Network*. In Proc. ACM SIGCOMM, 2001.

[19] Rham, E., Bernstein, P. *A survey of approaches to automatic schema matching*. The VLDB Journal 10: 334-350 (2001).

[20] Rowstran A. and Druschel P. *Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems*. In Proc. of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware) 2001

[21] Mediator : LeSelect : `http://www-caravel.inria.fr/leselect/`

[22] Stoica, I. et al. *Chord: a Scalable Peer-to-Peer Lookup Service for Internet Applications*. In Proc. ACM SIGCOMM Conf., 2001.

[23] Tatarinov, I. et al. *The Piazza Peer Data Management Project*. ACM SIGMOD Record, 32(3), 2003.

[24] Yang, B., Garcia-Molina, H. *Designing a Super-Peer Network*. In Proc. of ICDE, 2003