Content Based Clustering for Semantic P2P Data Integration

Ahmed Moujane, Dalila Chiadmi, Laila Benhlima, FaouziaWadjinny
Laboratoire SIR
EcoleMohammadiad'Ingénieurs
Université Mohammed 5 - Agdal Rabat
Avenue Ibn-Sina B.P. 765 Agdal Rabat Morocco
{Moujane, Chiadmi, Benhlima, Wadjinny}@emi.ac.ma

ABSTRACT: *Clustering peers based on the semantics of their content is one of the most difficult and important taskin P2P data integration systems because it enhances data search and integration significantly. Currently super-peer networks, such as the Edutella network, do not provide sophisticated means for such a "semantic clustering" of peers.In fact, most solutions try only to combine the advantages of data integration and P2P technologies to overcome centralized solutions shortcomings without taking into account content based semantic and efficiency to deal with network dynamicity. In this paper, we investigate P2P computing and data integration fundamentals and detail the challenges that face the P2P data integration process. In addition, we presentour approach for semantic clustering in a super peer architecture based on a vector space model forour P2P semantic data integration framework. In a first stage, we detail the various modules of our framework and specify the functions of each one. Then, we detailed the different levels of knowledge we take into consideration for the semantic clustering that we adopted using ontology alignment techniques. Finally, we explain how we manage network dynamicity and how semantic should be adjusted accordingly.*

Keywords: Semantic Integration, P2P, Ontology Alignment, OWL, Semantic Clustering

Received: 15 September 2011, Revised 22 November 2011, Accepted 1 December 2011

1. Introduction

In P2P data integration systems,clustering peers based on the semantics of their content is one of the toughest and the most importanttask because it enhances data search and integration significantly. Actually, P2P systemsoffer more and moreonline sourcesand accessibility thanks to the Web and Network infra-structures advances.In such large scale systems, searching information is a hard effort for usersthat could be flooded by the amount of distributed information and systems they access to. Grouping information by domains, in "small" clusters, would significantly enhance and optimize the information searching process. So, theapproach of clustering must beefficiently accomplished based on the content semantic of the data sources to be integrated and which are often heterogeneous. Unfortunately,semanticheterogeneity is identified as one of the hard and essential problems when dealing with interoperability of multiples data sources [1]. Super-peer networks, such as the Edutella network, do not provide sophisticated means for such a "semantic clustering" of peers. In fact, most solutions try only to combine the advantages of data integration and P2P technologies to overcome centralized solutions shortcomings without taking into account content based semantic and efficiency. Effectively, several P2P data integration works are proposed, but most of them are mediator-based and do not really cope with distribution. Indeed, they suffer from static matching as done in mediator architectures,which are not adequate in the case of peer-to-peer based systems consisting of a set of completely autonomous

peers in the definition of their data and knowledge. Indeed, a peer is both a client and a server, and should cooperate with other peers in a transparent way depending on the capabilities previously concluded withouthaving any global schema or kind of centralized knowledge due to the large scale and network dynamicity.

In this paper, we investigate **P2P** computing and data integration fundamentals. Then, we study the problem of data integration in the P2P setting: How can P2P techniques help data integration in distributed environment? What new issues and challenges should be surmounted? Then, we present our solution for P2P semantic data integration for which we will detail the content based semantic clustering process in a super peer architecture built on a vector space model. Then, we present our approach for network dynamicity management.

2. Data Integration

Data Integration aims to query heterogeneous, autonomous and distributed data sources. Different kinds of heterogeneities are identified. According to this, we distinguish three levels of data integration.

- Syntactic integration intends to overcome heterogeneities that come from the use of different data types and models to structure the same information (Relation in relational databases, classes in OO databases, XML tags, etc.).

- Structural integration has for task to resolve heterogeneities that rise from using different schemas to represent the same data. They are closely related to the choices of conception. For example, the name of a person can be represented by only one field "name" or by two fields "*fisrt_name*" and "*last_name*".

- Semantic integration is by far the most complicated task. It aims to resolve heterogeneities that outcomes from the differences of significance, interpretation or use of the same data. For instance, (1) Words that have similar meaning but havedifferent forms (e.g. Book and manuscript) will not match each other; (2) Words having multiple meaning (e.g. java) will make confusion and brings noisy responses.

To defeat these heterogeneities and provide access to heterogeneous and distributed data sources, several works and studies from the early 80's have been proposed. In this context, solutions using conceptual models, such as relational models, XML and ontologies were adopted. For example, we mention systems like Information Manifold [2], TSIMMIS [6], WASSIT ([3], [4], [5]). These classical centralized approaches are based on the supposition that it is possible to define a virtual global schema using approaches like GAV [6] and LAV [2, 27]. These approaches serve to establish mappings between the global schema and the local schemas. User's queries are then posed on the global schema to be translated, rewritten and optimized in further stages. However, most of these solutions suffer from client-server drawbacks.For this reason, we should take advantages from the P2P computing techniques.

3. P2P Computing

"*Peer to peer*" is a set of techniques permitting to build a data sharing system among several users. As pointed out by Shirky[7], P2Pis "*A class of applications that take advantage of resources-storage, CPU cycles, content, human presence- available at the edges of the Internet*". The birth of P2P networks is one of the events that have marked the Internet growth in the current decade. The P2P flux percentage, in relation to the Internet communication traffic, has been estimated, by the Cache Logic P2P technology company, to be between 64% and 84% for the year 2005. No statistics can ever be entirely truthful on the internet traffic but this is a really good estimation. Classical P2P networks have become recently a great infrastructure. Systems like Napster, Gnutella, Kazaa have showed their exploits for file sharing. Indeed, file sharing was the main incentive for several of these successful platforms. However, most of them focus on handling semantic-free items and support only key-based and in some cases keyword-based search. Doing so restricts classical P2P networks convenience and limits the techniques that can be employed for distributed data.

With the growth of P2P as an alternative for Client/Server systems, several works and architectures have been presented about different aspects of P2P computing. Any P2P computing application relays on a logical network, called overlay network, composed of a set of nodes and links between them that form logical connections. This overlay network is virtual and has its architecture based on a physical IP network. We can classify this P2P architecture depending on two criteria: decentralization and structure. We present these two points in the next sections.

3.1 Decentralization

Theoretically, overlay networks are supposed to be purely decentralized. However, this is not always true in practice and

different degrees of decentralization are met. Generally, we can distinguish three architectures of decentralization: pure, hybrid and super peer or partial architecture. In the totally distributed architecture, all nodes are equivalents and play the same role. There is no central control or coordination of their tasks. Each node acts as both a client and a server. There are no relaying services and the whole communication exchange process is accomplished directly between nodes. In the hybrid centralized architecture, there is at least one central point of control. All peers store their indices in the central server or cluster of servers. Peer localization and reference are done by the global control servers.The data transfer is accomplished in an end-to-end way directly between peers. This architecture has client-server-like drawbacks. Finally, in the partially centralized architecture, some nodes called super peershave roles that are more important. Each super peer contains indexes of files contained in the subscribed peers. The super peers and the nodes that it subscribes constitute a cluster. From the outside of a cluster, the communication with each node within this cluster must be initiated through its super node. The way the super node is designed by the network varies between systems. We must note that the set of super nodes would not be a point of failure because the designation is dynamic. In case of failure, a new super peer is designed for the cluster concerned. Thus, super peer architecture offers more advantages than the other ones for P2P data integration.

3.2 Structure

Network structure refers to where and how nodes and data are added to the system. In-deed, the overlay network can be created in an undetermined way (ad hoc) as nodes and content are added, or whether based on specific policies. We classify peer-to-peer structure in two classes, which are structured and unstructured. The two classes are described below.

3.2.1 Structured

Structured P2P networks are characterized by the use of identifiers assigned to data items. The overlay network organizes its nodes into a graph that maps each data identifier to a peer offering a highly controlled logical topology. This mapping between content and location is accomplished through a form of distributed routing table permitting queries to be efficiently routed to the peer with the desired data. In this way, structured overlay networks can partially resolve scalability issues encountered in P2P environment and rare items are easily located. However, it is hard to maintain the network structure for efficient routing process due to the volatile nature of P2P systems. Representative systems of Structured P2P overlay networks include Content Addressable Network (CAN) [8], Tapestry [9], Chord [10], Pastry [11] and Kademlia [12].

3.2.2 Unstructured

In unstructured networks, peers join the system without any prior knowledge of the network topology. Thus, data files are located independently from the overlay topology. The location of files is not restricted or founded on any network knowledge. This makes data locating a hard task in systems that offer query functionalities. Search mechanisms are based on a large spectrum of techniques varying from basic methods such as flooding the network by propagating the queries, to more sophisticated techniques that uses routing indexes and random walks. Flooding-based techniques are useful for locating highly replicated items and are flexible for peer joining and leaving but they are less adequate for locating rare data objects. Obviously, those techniques make unstructured systems faced to scalability and availability issues because peers are quickly overloaded as the systems size and the number of queries increase. On the other hand, unstructured networks are accommodated for highly volatile systems where nodes are joining and leaving at high rate. Among unstructured networks, we find Napster, Kazaa, and Edutella.

4. P2P Data Integration Issues

Data Integration issues were widely investigated in the literature [13], [14]. Several centralized solutions based on the assumption that it's possible to define a virtual global schema using approaches like GAV and LAV were proposed. However, in the P2P context, this assumption does not hold anymore due to the highly distributed and dynamic nature of the P2P infrastructure. Each peer has a limited view on the system, and cannot give complete answers. Therefore, data integration brings and raises new challenges in the P2P environment. Many P2P solutions have been proposed to resolve some of these issues using different techniques and concepts. We can distinguish two main categories, PDMS (peer data management system) and OPDMS (ontology-based PDMS). The first family is characterized by the combination of schema-based integration techniques and P2P infrastructure. Among solutions that we classify in this family, we find LRM Model [15], PIAZZA [16]. The second category arises by combining PDMS techniques and the use of ontologies that has been acknowledged to be an efficient approach to advance interoperation of distributed data sources at a semantic level [17], [14]. Among solutions that belong to this second family, we can mention SWAP architecture and those founded on it like the Bibster system [18]. P2P ontology mappings and query processing are two main issues in an OPDMS. Ontologies are used in local peers as a conceptual schema of the underlying

information. Nevertheless, the use of ontologies in a P2P setting creates new issues because we have to deal with more than one ontology. In fact, ontologies have been lately introduced for knowledge representation in information systems. Indeed, resources are generally described by XML schemas, relational schemas, etc. Even in the case where they are described by ontologies, their representation doesn't adopt the same standards and do not use the same languages (OWL, DAML+OIL, RDFS, etc.) [4]. In most cases, the answering Peer does not know all the terms used in the query expressed according to the local ontology of the requesting peer. Thus, to solve this problem, a mapping between both requesting and answering peer local ontologies must be accomplished. Due to these mapping challenges and incomplete knowledge, it is unfeasible to assure efficient query processing without exhaustive research, which is inconceivable in large-scaled P2P systems. So, query routing, processing and optimization techniques should be considered.

However, Query processing and routing in a PDMS is different from querying in flat P2P system. In fact, in PDMS, indexes take the form of schemas and we do not have hash functions or global indexes that could permit to find the requested data because of the presence of heterogeneous schemas at the peers. In contrast, the originality of PDMS lies in its ability to exploit the transitive relationships among such schemas. Thus, efficient query processing should generate distributed query plans according to the information returned by the routing process. The query plans are calculated according to the cost of alternative operator order (e.g., join/ aggregation reordering) and execution strategies (e.g., data or query shipping) for these plans. The query plans are then executed by the selected peers according to their capabilities.

PDMS policies to deal with query processing are diversified but most of them are inspired from the classical P2P world. Edutella[19] uses routing indices [20] for their super-peers providing RDFS schemas and RDF metadata of their description. Queries are based on a Datalog-based query exchange language called RDF-QEL to serve as a common query interchange format. Systems like PeerDB[21] uses flooding and information retrieval (IR) approaches to decide, at the query run-time, the mappings between two peer schemas. In PeerDB, no predefined mapping tables are established, only descriptive metadata are associated to schema attributes. Query routing is accomplished through flooding neighbor's peers which uses IR techniques to decide if matching attributes for the query exist in the local schema. Then, the user based on the returned suggestions, must choose which queries are relevant and which ones can be executed. The system use caching mechanisms for the user selected preferences to route future similar queries. Piazza [16] utilizes a declarative XQuery-based mapping language to define the mappings established between peers of individual peer schemas. Based on the chains of mappings between peers of nodes, a reformulation algorithm produces query expressions equivalent to a given query; that can be routed and distributed across the Piazza network. Like Piazza, Hyperion [22] uses mapping tables and mapping expressions (mapping tables permitting variables) to define correspondences between local schemas in peers. Based on the mapping tables and mapping expressions, a query manager rewrites queries posed in terms of the local schema, in order to be processed by the acquainted peers.

5. Our Solution

Racall that, in the light of having investigated the mechanisms for data integration and techniques for P2P computing, we have explored issues for P2P data integration and querying in peer-to-peer data management systems. Based on this exploration, we now focus on presenting our solution for semantic P2P data integration framework that is based on our platform WASSIT (*frameWorkd'intégrAtion de reSSources de donnéesfondéesur la médIaTion*) for central data integration. In this paper, we present our architecture and detail each of its components. Then, we give an overview for our knowledge representation. Afterward, we present our approach to manager network dynamicity and volatility.

Our work is motivated by the need to integrate several heterogeneous data sources and systems that belongs to different governmental institutions. Each system can be seen as an autonomous system managed independently and no global schema can be conceived. These systems hold semantically equivalent our related information of different domains.Due to the large scale of the systems and data, and financial constraints, centralized solutions like WASSIT or TSIMMIS [6] wouldn't do the job. Consequently, the recourse to both P2P and data integration technologies seems to bepromising for an efficient solution. For feasibility raisons, we will implement our solution for institutional digital libraries as a case study.

5.1 Architecture

We have adopted, for our framework, an unstructured super peer architecture where each node can be a peer or a super peer (Figure 1). A super peer and the peers that are connected to it constitute a cluster. The super peer is responsible for lookup with the network for its cluster. Each node should contact its super peer to submit user queries out of the cluster. Each peer that

hopes to join the P2P network should have an entry point. Through its entry point, a new peer should provide expertise about the information resources that it manages. Then, based on the expertise, each peer is registered by a super peer.

For us, each node represents an autonomous information system, and data integration is accomplished by establishing mappings among the various peers.

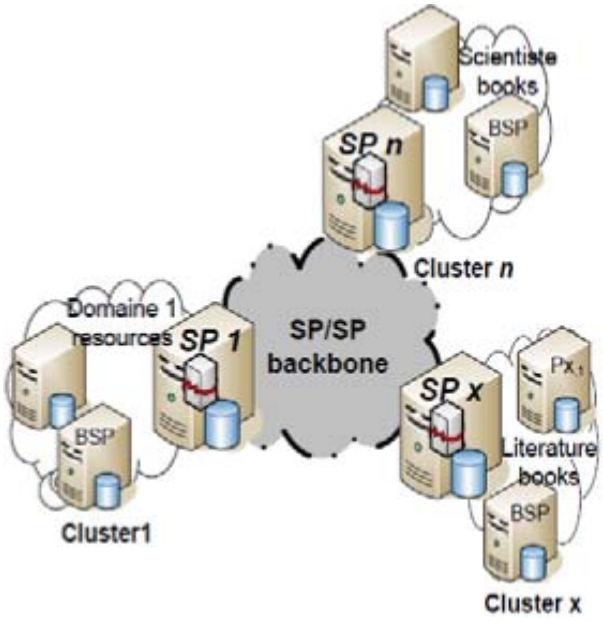

Figure 1. Our P2P network scenario

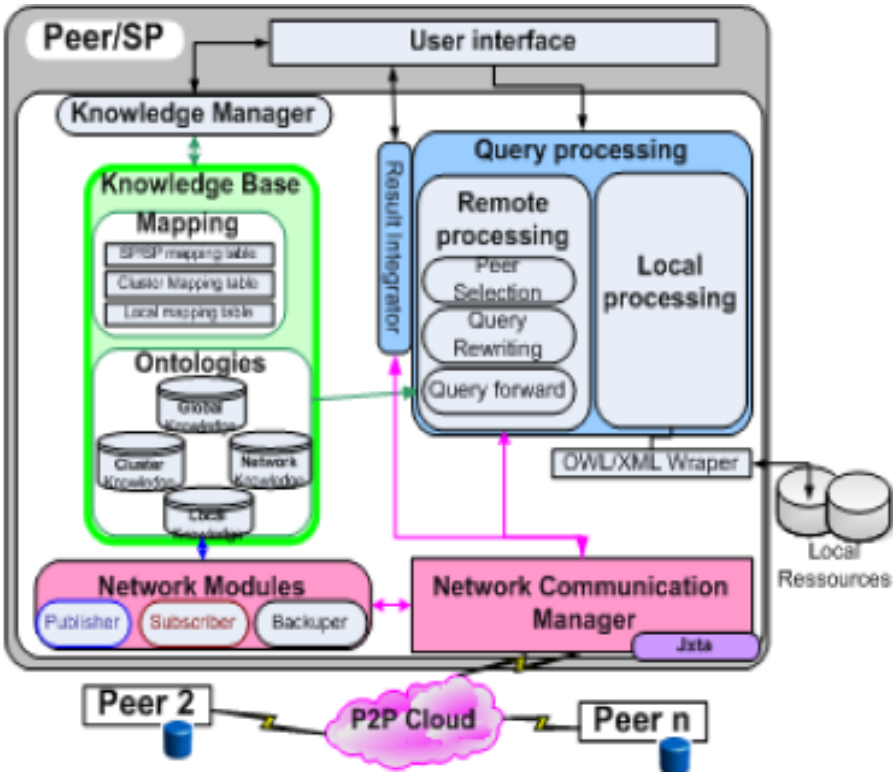

Figure 2. Our functional architecture

We introduce a new concept that we called Backup Super Peer (*BSP*) whose role is to ensure a passive backup for the super peer. The *BSP* is different from the k-redundant super peer that ensures an active backup or load balancing. The *BSP* is elected among peers that have enough hardware and network resources and it contains an image of the super peer knowledge.

In a first stage, our solution will be adopted for digital libraries context. However, it is intended to be a general framework. The architecture of our solution is presented in the figure 2 and contains the modules described below.

***Query processing*:** It is a coordinating module managing the process of query resolution. It receives queries from the user interface or from remote peers. Either way, it undertakes to answer the query locally and/or forward it to remote peers depending on the query specifications. The routing decision to which peers a query should be forwarded is based on the knowledge about other peers. This module is composed of two sub-modules. The first one is dedicated for local processing and interacts directly with the wrappers of local resources as it is detailed in the [23] for the WASSIT platform. The second sub-module accomplishes remote query processing. It has for mission peer selection, query rewriting and query distribution for remote processing.

***The network module*:** it has for task to manage peer joining to the network by choosing the pertinent cluster. The joining process is based on the available knowledge at the peer and on the network. Each peer that wants to join the P2P network should submit through its entry point a TTL-query containing description of its local knowledge using a summary of the most heavy-weighted concepts M_c and a threshold S_S for semantic similarity to satisfy by the available super peers. This mechanism is used to avoid network congestion and resources waste. Each super peer SP_X that receives the subscribing query of the new peer P_{new}, calculates the semantic similarity (O_{CL,SP_x} , $O_{LS,P_{new}}$) between its cluster knowledge and the asking peer description. If the semantic similarity is over the specified threshold S_S than the super peer sends an acceptation response containing the calculated similarity value to the asking peer P_{new}. Among the returned values, the query initiating peer chooses the better cluster to join. The decision is taken by the initiating peer to avoid inter-super peer conflict and inter-blockage.The network modules have also for task to accomplish another role of backuping in the case of the *BSP* nodes. The backup consists of backuping the super peer knowledge, providing a proactive rescue and preventing reactive election in the case of super peer failure.

The knowledge component differs depending on the nature of the node. In the case of the peer nodes, the knowledge component contains local expertise and information about cluster knowledge. However, for the *SP and BSP* the knowledge component contains also the network expertise. The knowledge contained in this component is represented by ontologies and mapping tables that we describe in section 5.3.

User interface allows users to formulate and submit queries. It is also used by the peer's administrators to manage resources and to define constraints and mapping.

Network communication manager is a component that is responsible for network communication between peers. It provides transport mechanisms to hide low-level communication details for the other components of the system, while sending and forwarding queries. In a first stage, we have opted to use the JXTA system as the communication platform for the implementation of our framework. Thus, the peers will be identified by the JXTA ID URN in the underlying P2P infrastructure.

5.2 Architecture Formalization

Formally, we present our PDM as a couple (G, M) where

$G = \{(SP \cup P), (Eg_{SP\text{-}SP} \cup Eg_{SP\text{-}P})\}$ is a labeled non-directed graph where:

$SP = \{SP_1, SP_2, ..., SP_n\}$ is a set of super-peers, each one with its ontology and expertise

$P = \{(P_1, P_2, \ldots, P_m)\}$ is a set of peers each one with its own ontology and expertise

$Eg_{SP\text{-}SP} = \{(SP_i, SP_j) \mid SP_i, SP_j \in SP_2\}$ is a set of semantic links between Super-peers inside the backbone network.

$Eg_{SP\text{-}P} = \{(SP_i, P_j) \mid n\ SP_i \in SP, P_j \in P\}$ is a set of semantic links between peers and super peer inside each cluster.

$M = \{M^{ij}_{SP\text{-}SP}\} \cup \{M^{kl}_{SP\text{-}P}\}$ is a set of correspondence or alignment matrix, each one is associated with a semantic link $Ls_{i,j} \in Eg_{SP\text{-}SP}$, $Ls_{l,k} \in Eg_{SP\text{-}P}$ containing alignment information between the peers ontologies and those of the superpeers of a cluster for the $M^{ij}_{SP\text{-}P}$ matrix and the semantic alignment between superpeers representing different clusters for the $M^{ij}_{SP\text{-}SP}$

matrix.

5.3 Knowledge Representation

One of the major challenges in P2P data integration is the definition of the knowledge and semantic to share across the network. Once this semantic is specified, we should define the format of its representation, the way of semantic construction (manual, automatic or semi-automatic), the semantic that could be shared between peers and between superpeers and how it is distributed across the whole system.

5.3.1 Data Model

In each node, sources are indexed using a semantic vector that describes the weight of concepts calculated using TF-IDF or similar schemas. These weighted indexes and schemas are used to construct local ontologies. The ontology construction is not our principal goal in this works. We suppose that local ontologies are already established. For simplicity raisons, we define a space model for ontologies, we denote W this ontology space. Each ontology $O \in W$ is defined as a four-tuple $<C, R, AC, AR>$ where C is a set of unary predicates called concepts, $R \subseteq C \times C$ is a set of binary relations called properties and AC are AR are axioms for concepts and relations respectively. The definition of an otology space is dictated by necessity to use computational operations for ontology comparison, mapping, alignment and merging or for similarity measurement.

To be simpler, the ontology space is projected on a Vector Space Model (VSM) in which each ontology is represented by a vector V_O in this space where the concepts are the dimensions. In the same manner, a user query is represented as a query vector V_Q and the similarity between the user query and the target peer ontology is estimated by the distance between V_O and V_Q. In this way, we merge the problem of semantic-based search with query routing in our overlay network.

5.3.2 Data Model Pivot Language

As presented in our framework for centralized data integration, WASSIT we have chosen to use OWL ontologies to represent the semantic knowledge. The benefits of using OWL ontology are not to be demonstrated in this context. Indeed, in [4] we have presented how to build semantic knowledge using semi-automatic mapping from XML-documents to owl ontologies. We have also developed and tested a Wrapper from relational to XML documents in order to map theme into OWL ontologies[24].

XQuery [25] is the query language we adopt in our platform since it is the W3C standard for querying XML documents. In order to achieve efficient query processing, we represent the queries according to an algebraic model. We have chosen to use XAT (XML Algebra Tree) as algebra for query representation in our framework.

As our P2P integration uses semantic clustering based on peer's content, we have introduced the notion of expertise for each peer.This notion is used during the stage of peer's clustering. Each node that hops to get into the network should publish a summary called also expertise of its data sources and capabilities. The expertise notion is used to optimize the clustering process by preventing heavy computations of calculating similarity measuresbetween the holes ontologies. In this way, the peer expertise is used as a light weighted ontology containing the potentially most important concepts,relations and a description of the topics and subjects that the data sources are interested in. The expertise notion can be seen as a kind of the domain ontology enclosing complement of information about the data sources. It contains information that is not expressed by the local ontology describing the schema level of data sources (XMLschema, attributes of relational tables).For freshness reasons, we introduce in our data model a freshness parameter which will be incremented after update of the ontology published by each node. The idea of introducing freshness is to keep cluster and network ontologies up to date regarding to the peer knowledge. This parameter permits to elaborate an infrastructure permitting to manage knowledge freshness depending on the network dynamicity. To this end, we use *owl:priorVersion* and *owl:versionInfo* constructs to implement the freshness parameter.

5.3.3 Expertise

As presented before, each peer that hopes to be connected to the network must publish an expertise describing information and capabilities about its data sources.As shown in the figure 3, the expertise is a supplement of information about the *type* of resources (book) and the *topic* of the books contained In the XML document. In this example, we can see that P1 contain books written in English languageconcerning Smartphones and the Android Operating System.This information is not included neither in the XML document nor in the XML schema published as an ontology. The expertise is used during the clustering process. Indeed, sources that concern informatics data are grouped in the cluster of informatics. And sources that concernmedicalresourcesare grouped in the medical cluster.

In this case, we distinguish four levels of information and data: The first one is the information of the XML schema. The second

one is the information contained in the XML document. The third level is the resourceitself, in our case of digital libraries, whichis the books (paper, e-book,articles, etc.) to be read, downloaded or bought. And the forth level is what we called expertise. This level contains information about the topics of the books, services offered (online read, download, translation, etc.).

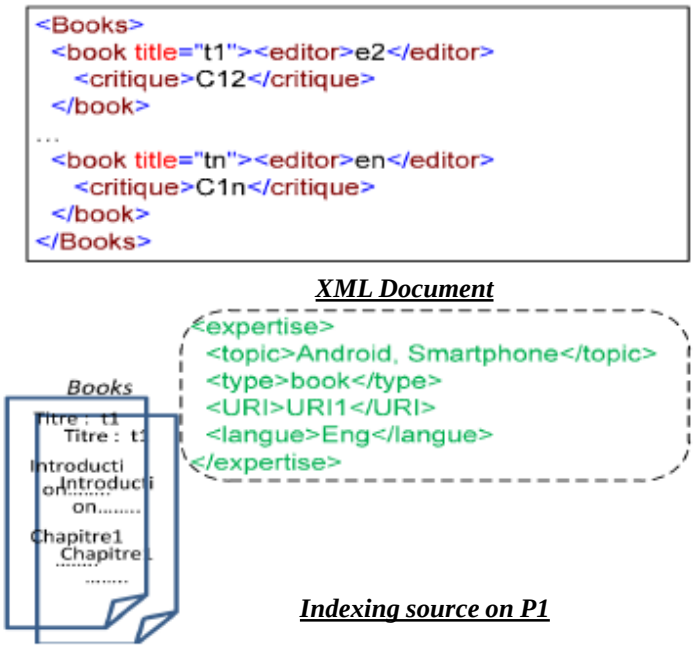

Figure 3. Information levels of peers

5.3.4 Ontology

In addition to the expertise, each peer publishes an ontology describing itsresources. The ontology contains concepts of the schema level of the data sources. This ontology defines constraints about the XML Schemas and relational databases attributes (cardinalities, types of fields, etc.), semantic relationships between the concepts and properties, etc. The ontology describes the concepts and terms, concerning the application domain (digital libraries, e-learning, etc.) that the system is conceived for.

5.3.4.1 Ontology Concepts and Properties

In our OWL ontology, we consider all concepts and relation between themes. For the relations, we distinguish two types of relations between nodes:

- The first one are Relations defined by the users or administrators using *OWL:objetProperty* construct. For instance, the "Is_the_author_of" relation in the example below:

<owl:ObjectPropertyrdf:ID="Is_the_author_of">

<rdfs:domainrdf:resource="#Author"/>

<rdfs:rangerdf:resource="#Book"/>

</owl:ObjectProperty>

- The second one is are OWL natives semantic and hierarchical implicit relations such as equivalence (*owl:equivalentClass*, *owl:equivalentProperty*), specialization and generalization (*rdfs:subClassOf*, *rdfs:subPropertyOf*), etc. For simplicity raisons, in a fist stage, *owl:DataTyeProperty* are not taken into account.

Our ontologies are assumed to be described in a language like OWL-DL for raisons of decidability and reasoning. We distinguish four types of ontologies, the local ontology O_{Ls} that represents the knowledge of local resources of each node. The second one represents the ontology contained in a cluster $O_{Cl\text{-}SPx}$ and is shared between peers of the cluster. The network ontology $O_{Net\text{-}SPx}$ is hosted by each super peer and describes the knowledge out of the local cluster as seen by the super peer SP_x. These three ontologies will be merged to obtain the global ontology $O_{g\text{-}SPx}$ of the super peer SP_x.

5.3.5 Mapping Tables

We distinguish three types of mapping tables. The basic one MJ_{Ls} is used locally in each peer to assure it as an autonomous information system. The second mapping table $MJ_{Cl\ -\ SPx}$ is between peers and their super peer. The third one $MJ_{Net\text{-}SPx}$ is between Super peers of different clusters.

Mapping tables are presented as matrix of semantic vectors SV where lines represent concepts and columns represent peers for the $MJ_{Cl\ -\ SPx}$and clusters for the $MJ_{Net\ -\ SPx}$.

Our mapping tables are extracted based on the alignmentprocess presented in section.

5.4 Content based Semantic Clustering

Recall that our system is based on a superpeer architecture in which clusters are built based on the sources content semantic. A new peer P_{new} that hopes to join the network send à join request containing its expertise.Each superpeer that receives the joinquery calculates a similarity between its expertise (expertise of the cluster) and that of the P_{new}. The calculated similarities are sent to the requestor peer. The later chose the most similar cluster depending on the topics that it is interested in. Note that a peer can belong to more than one cluster, one per domain if it is interested to several domains. This choice is made to assure completeness of user query answers.

The similarity calculation is based on the expertise information. As we have presentedearlier, the expertise can be seen as a light ontology. So, the comparison is based on approaches inspired from ontology alignment techniques. So, we distinguish the similarity comparison based on the terminological, structural and semantic levels.

5.4.1 Terminological Similarity

Terminological similarity is calculated based on the Edit distance. Each terms (concept) contained in the expertise of a new peer P_{new} is compared terminologically to theconcepts of cluster expertise as proposed in [stab] using the equation (1).

$$SM(L_i, L_j) = \max(0, \frac{\min(|L_i|, |L_j|) - ed(L_i, L_j)}{\min(|L_i|, |L_j|)}) \in [0,1] \quad (1)$$

5.4.2 Structural and semanticsimilarity

Given the expertise of the P_{new} and that of a cluster that receives the join query. Once the super peer P_{new} expertise, for each couple (C, C')$\in$ exp(P_{new}) exp(cluster), a structural similarity using an externaldomain ontology is calculated. We use theWu–Palmer similarity measure[29].

$$Sim_{WP}(C_1, C_2) = \frac{2 * depth(LCA(C_1, C_2))}{depth(C_1) + depth(C_2)} \quad (2)$$

The LCA (c1,c2) is the Lowest Common Ancestor of the concepts (c1,c2), and Depth(c) is the depth of the concept c in relation to the root.

5.4.3 Global Similarity

To calculate the global similarity between the expertise of the two expertise of (P_{new} and of the cluster), we use the TVERSKY similaritymeasure (equation 3). Depending on this measure, the similarity between two sets of elements A and B, iscalculated based on their common attributesand those of their difference. The most common attributes the most they are similar and vice versa.

$$Sim(a,b) = \frac{|A \cap B|}{|A \cap B| + \beta |A \setminus B| + (1-\beta)|B \setminus A|} \quad (3)$$

Where A$\cap$ B is objects that are in both A and B, A\B are the objects that are in A and not in B, and B\A are the objects that are

in B and not in A. β is a coefficient that gives importance to non-common objects of A or B.

As we are interested in semantic similarity and not in exact equality of objects, we havebeen inspired from fuzzy logic theoretic to deal with the concept of partial truth(partial similarity), where the truth value may range between completely true (similarity = 1) and completely false (similarity = 0).

So, our two fuzzy sets are: the P_{new} expertise concepts set and the cluster expertise concepts set. So the intersection and differences of objects are exact but depending on the found similarity (Terminological SM + structuralsimilaritySim_{WP}).

5.5 Ontology Alignment

In this section, we presentthe ontology alignment process for our framework. As showed before, each peer publishes an ontology describing concepts of the schema level.After having chosen the cluster to witch a peer will belong, the ontology of this peer is aligned with the cluster ontology in order to merge theme in a new ontology of the cluster. The new cluster ontology encompasses all the concepts of the cluster peers. To accomplish the process of alignment, we distinguish three levels ofsimilarity terminological, structural and semantic for our OWL ontology entities. Before givingthealignmentapproach for our solution, we present the vector space model we adopted to represent our ontology.

5.5.1 Vector Space Model

(a) Graph ontology

Before analyzing any relation type, we should present our owl ontology as a semantic graph where the nodes are concepts and the edges are the semantic relations between the concepts to be taken in semantic analysis. As defined in [26], we define an ontology graph as below.

Definition: Given an ontology O, the ontology graph $G_O = (V, E, l_V, l_E)$ of O is a directed labeled graph. V is a set of nodes representing all concepts in O. E is a set of directed edges representing all relations in O. l_V and l_E are labeling functions on V and E, respectively. $e_{s,t}$ is the edge connecting ontology nodes s and t, where edge e belongs to edges set E, node s and t belong to nodes set V, we denote by $e \in E$, $s.t \in V$.

As we are interested in the semantic of the two types of relationsdetailed before, our ontology graph will be represented in a vector space model with a matrix $n \times n \times k$, where n is the number of concepts V and k is the number of types of relations E. So, our graph ontology can be represented as bellow

$$M = \begin{pmatrix} (d_1, d_2, ...d_k)_{11} & (d_1, d_2, ...d_k)_{12} & (d_1, d_2, ...d_k)_{1n} \\ (d_1, d_2, ...d_k)_{21} & (d_1, d_2, ...d_k)_{22} & (d_1, d_2, ...d_k)_{2n} \\ (d_1, d_2, ...d_k)_{n1} & (d_1, d_2, ...d_k)_{n2} & (d_1, d_2, ...d_k)_{nn} \end{pmatrix}$$

Where $m_{c,i} = (d_1, d_2, .. , d_k)_{c,i}$ represents the relationship between the concepts *C* and I based on the K dimensions we take into account. It defines also similarity correlation between concept *C* and I.

The Concept C is represented by the vector of vectors: $V_c = (m_{c,1}, m_{c,2}, \ldots, m_{c,n})$

The Similarity between two concepts X and Y is equal to the similarity between their vectors in our vector spacemodel. It is calculated by the scalar product between the two vectors V_x and V_y which we denote X and Y:

$$S'_{X,Y} = X \times Y = \sum_{i=1}^{k} \sum_{j=1}^{n} X_{i,j} \times Y_{i,j}$$

As we defined the similarity function in the interval [0..1], the similarity is normalized by the equation bellow:

$$S_{X,Y} = \frac{S'_{X,Y}}{\| \sum_{X \in O} \sum_{Y \in O} X \times Y \|}$$

(b) Alignment of Ontologies

As mentioned earlier, the ontologies of the nodes in each cluster must be combined in a one global ontology for this cluster.To builda suchontology,we adopt an ontology merging process.We first establish an alignment of the ontologies of each new peer and the existing ontology of the cluster. Our alignment process must come up with various aspects of the semantic of the ontologies terminological, structural, and semantic.

We have each ontology represented as a graph. So, the alignment of ontologies can be seen as a problem of graph matching. Thus, to calculate the alignment between two OWL ontologies represented in our vector space model, we have applied the graph matching presented in [31]. The applied algorithm calculates the similarity between entities in the OWL ontology graph by iterating recursivelythe following equation.

$$S_{L+1} = B*S_L*A^T + B^T*S_L*A, L= 0, 1, 2 \quad (4)$$

Where S_L is a $N_B * N_A$ similarity matrix of entries s_{ij} at the iteration *k*, and A and B are $N_B*N_B*N_d$ and $N_A* N_A* N_d$ three-dimensional matrices representingthe graphs G_A respectively G_B.N_A and N_B are the number of rows of A and B, and *d* is the number of predicates selected as dimensions of the VSM.

As done in [??], the similarity matrix S0 is initially set to 1 (supposing that all entities from G_A are initially equal to all entities in G_B) for the dimensions for which we don't have any similarity measure. If we have already known similarity values of some pair of entities, we can adapt this matrix accordingly, and keep the known values.

Example:

Let's take a simple example where we have two ontology directed graphs G_A and G_B as shown in Figure 4we have three entities in each directed graph and two dimensions d_1 and d_2. A dimension is an owl semantic inherent relation (*rdfs:subClassOf, rdfs:range, rdfs:domaine, etc*.).

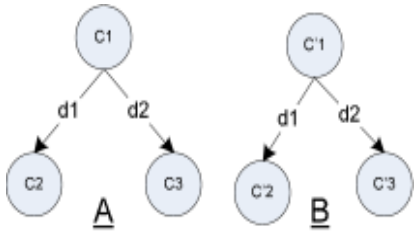

Figure 4. Information levels of peers

$$A = \begin{pmatrix} (0,0) & (1,0) & (0,0) \\ (0,0) & (0,0) & (0,0) \\ (0,0) & (0,0) & (0,0) \end{pmatrix} B = \begin{pmatrix} (0,0) & (1,0) & (0,0) \\ (0,0) & (0,0) & (0,0) \\ (0,0) & (0,0) & (0,0) \end{pmatrix} S_0 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

$$S_1 = B * S_0 * A^T + B^T * S_0 * A = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

This operation is iterated until S_L values converge. To normalize the similarity matrix (to retain its values in the interval [0, 1]), we can divideall its elements by the Frobenius norm of the matrix defined as the square root of the sum of the absolute squares of its elements.

$$\| S \|_F = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} | s_{ij} |^2}$$

5.5.3 Dimensions of the Vector Space

To assure a significant alignment calculation we can just the following semantic dimensions that we motivate.

√ *rdfs:subClassOf, rdfs:subpropertyOf:* to take into consideration the taxonomical level of the ontology.

√ *rdfs:range et rdfs:domaine* pour to represent the internal, external relations (attributes, relation represented in owl respectively using *Owl.DataypeProperty* and *Owl:ObjetProperty*) and instances:

√ *Owl:equivalentClass, owl:equivalentProperty,* to take into consideration equivalence between classes respectively properties that can be defined differently.

√ *rdfs:label,* This dimension is used to calculate the similarity on the terminological level found identically as used in the case of the expertise presented earlier.

In order to have more precision on the ontology alignment, wecan add other ontology dimensions *owl:unionOf, owl:intersectionOf owl:complementOf,* etc.

Thegoal here is to present our idea without more details to avoid redundancies and because the paper remains self-contained. The alignment process was not so detailed, due to space limitation. However, more details will be presented in future works.

5.5.4 Related Works

As the object of our paperis not to present ontology alignment theoretical, we will present a brief overview of some works for finding structural similarities of graph entities. Indeed, some of them are developed precisely for ontology alignment while some others areadopted for other domains, like forWordNet similarity, but are still appropriate for the ontology alignment problem.

Initial works around structure-based semantic similarity focuses only on taxonomies (is-a constructs). Works similarto [28] measure the distance between the different nodes based on the path from one node to another. The shorter the path is, the more similar they are. Wu & Palmer measure calculates [29] calculate the path length to the root node from the least common subsumer (LCS) of the two entities, which is the most specific entity they share as an ancestor using the equation (2). In [30]the shortest path between two entities is found, and its value is scaled by the maximum path length in the "*is – a*" hierarchy in which they occur.

However, new ontologies are described using more expressive languages like OWL. Consequently, more sophisticated alignment approaches has been proposed recently.To match the particularities of new ontologies, [31] uses more sophisticated topological similarity measures, based on graph matching from discrete mathematics.

From it side, GMO is a bipartite graph matcher [32]thatbegin by considering the RDFrepresentation of the ontologies as a bipartite graph which is represented by itsadjacency matrix. The distance between the ontologies is representedby a distance similarity matrixcalculated based on the equation (4).

Our work has tried to take advantages form the previous work in order to adapt theme for our OWL ontologies.Once we have presented our approach for content based semantic clustering using alignment technics, we show in the following how we use this to deal with the network dynamicity management.

5.6 Network Dynamicity Management

To offer more flexibility to our solution, we have adopted an enhanced super peer model for which we describe the dynamic behavior under normal conditions. This model should cope with the situations of node joining, peer leaving, peer update, super peer leaving, *BSP* and *SP* election. In this section, we present how the semantic knowledge of each node is adjusted according to the network dynamicity.

5.6.1 Node joining

To join the network, a new node P_{new} should have an entry point P_{ent} that could be a peer or a super peer. If P_{new} has a predefined mapping $MAP\ (Q_{P\,new}, Q_{Pnent})$ to its entry point P_{ent}, which means that it has pre-defined idea on the network knowledge, then, P_{new} joins the cluster of its entry point and the mapping between its resources and the hosting super peer is accomplished using mapping composition $MAP\ (Q_{P\,new}, Q_{Cl\text{-}SPx}) = MAP\ (Q_{P\,new}, Q_{Pnent}) \otimes MAP\ (Q_{P\,new}, Q_{Cl\text{-}SPx})$. Otherwise, P_{new} sends a join request using TTL of J for the super peer of its entry point. This join request contains expertise metadata about P_{new} contents and computing resources, and is forwarded to super peer neighbors in the backbone network of J -hops. Every super peer that receives the join query will estimate if the querying node could semantically be hosted in its cluster or not, by calculating semantic similarities between the expertise of the cluster and that received from the P_{new}. In the affirmative case, the super peer compares its computing resources to those of the asking node. If the new node has greater computing power, it is prompted to be a new super peer for the cluster. But for more stability and to reduce computation and topology changes, we limit the clustering changes to only new peers that offer more significant power than the current super peer. A function for computing resources comparison should be defined based on the cluster size and the computing resources.

A new node that joins the network and has enough computing resources could build a new cluster if no super peer validates its join query. This case could happen if all clusters have reached the maximum size or if the new peer concerns a new topic.
Once a new peer P_{new} joins a cluster *fs* , a mapping between its knowledge and these of the hosting SP_x is calculated in an ascendant approach.

5.6.2 Peer leaving

When a peer P_L fails or simply leaves the network, its super peer would update its cluster knowledge accordingly. The changes are notified across the backbone network for neighbor's super peers. Peer could fail for many raisons such as network bottlenecks or maintenance problems. Peer failure is detected using keep-alive massages or when it does not re-play to queries. However, the procedure of leaving is initiated on demand by the asking peer using a disconnect request.

Once the peer departure is confirmed, the super peer adjusts the cluster knowledge $Q_{Cl\text{-}SPx}$ accordingly to obtain $Q'_{Cl\text{-}SPx}$. Our process for peer leaving follows a descendant approach and works as follows. For each concept C in the $Q_{Cl\text{-}SPx}$ ontology that refer to P_L in the mapping table $MJ_{Cl\text{-}SPx}$, if this concept refers to another peer $P_{I \neq L}$ then keep C else mark it for later deletion. A new mapping $MAP\ (Q'_{Cl\text{-}SPx}, Q_{Cl\text{-}SPx})$ between $Q'_{Cl\text{-}SPx}$ and the $Q_{Cl\text{-}SPx}$. It is then transmitted to the peers of the cluster to update their knowledge through mapping composition $MAP\ (Q'_{Cl\text{-}SPx}, Q_{LS\text{-}P}) = MAP\ (Q'_{Cl\text{-}SPx}, Q_{Cl\text{-}SPx}) \otimes M\ (Q'_{Cl\text{-}SPx}, Q_{LS\text{-}P})$. These changes are then forwarded out of the cluster X to update the Backbone overlay network.

5.6.3 Super peer leaving

As mentioned before, the super peer constitutes as single point of failure in super peer architectures. The super peer of a cluster could become unreachable due to failure or by simply leaving the network. In this case, the *BSP* proposed in our enhanced model will take the relief and prevents communication breakdown. No changes should be accomplished. Then, the *BSP* is converted to be the effective super peer and the cluster knowledge is updated accordingly. Afterward, the IP address and computing resources of the new super peer is communicated to the peers of the cluster and to the other super peers of the backbone network. Thus, super peer failure becomes as simple as any other node of the cluster.

5.6.4 The Backup Super Peer election

As mentioned, the BSP has for task to ensure fault tolerance in our enhanced super peer model. It is elected among cluster peer's that have enough hardware resources and offers more stability by being alive for long periods. This is accomplished throw historical activities. The election procedure is done in proactive way under normal conditions. The *BSP* continue polling periodically the *SP* to ensure the stability of the cluster. Moreover, a peer that could not communicate with its *SP* could query the *BSP* about the *SP* availability. Synchronization between the *SP* and *BSP* knowledge bases is accomplished periodically to ensure fault tolerance and self-organizing functionalities. The *BSP* knowledge is seen and processed as those of the other peers. Once the *BSP* is active, the knowledge of the old *SP* must be removed as explained for the peer leaving procedure. Then, the process of electing a new *BSP* is triggered.

5.6.5 Node update

It presents the most delicate procedure that can happen in P2P data integration systems. This is due to the fact that download

information by a peer should be shared in running time even before download accomplishment. However, fort simplicity raisons, we limit our approach to only totally downloaded documents. Thus, upon peer's knowledge changes, the cluster and network knowledge will be updated. To take into account these changes, two principal ideas can be adopted. The first and easiest one treats the peer as new one and deletes its old knowledge, and then uses the approach proposed in the node joining section. This idea do not reuse calculated mapping even more it should delete the old knowledge and re-compute the new knowledge and consequently, it is time and resources consuming. The second idea calculates a new mapping $M(Q'_{LS\text{-}Pm}, Q_{LS\text{-}Pm})$ between the old local ontology $Q_{LS\text{-}Pm}$ of the updated peer P_m and its update local otology $Q'_{LS\text{-}Pm}$. The new mapping is communicated to the super peer for composition of mapping to update cluster and global knowledge of the super peer SP_x. This second idea is better than the first one because it permits knowledge and mapping reuse.

6. Conclusion and Perspectives

Through the present paper, we have presented different aspects of P2P networks. The set of smart features of Peer-to-Peer architectures including decentralization, self-organization, scalability and fault tolerance offers significant advantages that can help semantic P2P data integration. Besides, we have showed that combining data integration techniques in a P2P context generates a multitude of challenges. Indeed, a peer has very little knowledge of its network and this information becomes obsolete very quickly. In addition, before worrying about how to process a query, a peer has to select which neighbor nodes should be queried. This decision is based on the knowledge that a peer has or shares across the system. As presented, knowledge representation and routing techniques are applied to deal with peer's selection.

Based on this study, we have developed our framework for semantic P2P data integration for which we have detailed the architecture and argued the conceptual choices. Besides, we have proposedour approach for content based semantic clustering in a vector space modelrepresenting the semantic knowledge. In addition we have presented an approach formanaging network dynamicity for our solution based on the WASSIT mediation framework.

We assume that our P2P framework has a plenty opportunities to support P2P semantic data integration and has the advantage to shun the scalability problem of most existing systems that uses centralized indexing, index flooding, or query flooding. To be competitive on a large number of PDMS, we must meet some other important challenges for optimization and efficiency. We are currently studying solutions for some of these P2P challenges and issues. Especially, we will focus in further works, on semantic query routing and processing. In parallel, we are working on implementing (similarity functions, alignment process and knowledge construction) our approach. To show the performance of our framework, experimental and simulation results will be published in future works.

References

[1] Moujane, A. D., Chiadmi., Benhlima, L. (2006). Semantic Mediation: An Ontology-Based Architecture Using Metadata, CSIT2006, Amman, Jordan, p. 317-326.

[2] Levy, A. Y., Rajaraman, A., Ordille, J. J. (1996). Query-Answering Algorithms for In-formation Agents. AAAI-96, Portland, Oregon.

[3] Benhlima, L. D., Chiadmi, Moujane, A. Gounbarek, L. (2007). Construction de la base de connaissances pour WASSIT . Journal marocain d'automatique, d'informatique et de traitement de signal. 3 (3).

[4] Moujane, A. D., Chiadmi., Benhlima, L. (2007). An approach for knowledge base con-struction, ICSSEA2007, Paris.

[5] Wadjinny, F., Chiadmi, D. (2008). Capacités de sources et optimisation de requêtes dans WASSIT, MCSEAI'08, Oran, Algeria.

[6] Chawathe, S. H., Garcia-Molina, Hammer, J., Ireland, K., Papakonstantinou, Y., Ullman, J. D., Widom, J. (1997). The TSIMMIS project: Integration of heterogeneous information sources, *Journal of Intelligent Information System*, 8 (2) 117–132.

[7] SHIRKY, C. (2000). What is p2p... and what isn't. Network, available online at http://www.oreillynet.com/pub/a/p2p/2000/11/24/shirky1-whatisp2p.html. O'Reilly.

[8] Ratnasamy, S., Francis, P., Handley, M., KARP, R. (2001). A scalable content-addressable network. *In:* Proceedings of SIGCOMM.

[9] Zhao, B., Kubiatowicz, J., Joseph, A. (2001). Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Tech. Rep. UCB/CSD-01-1141, Computer Science Division, University of California, Berkeley, 94720.

[10] Stoica, I., Morris, R., Karger, D., Kaashoek, M., Balakrishnan, H. (2001). Chord: A scalable peer-to-peer lookup service for internet applications. *In:* Proceedings of SIGCOMM.

[11] Rowstron, A., Druschel, P. (2001). Pastry: Scalable, distributed object location and rout-ing for large-scale peer-to-peer systems. *In:* Proceedings of IFIP/ACMMiddleware.Hei-delberg, Germany.
[12] Mayamounkov, P., Mazieres, D. (2002). Kademlia: A peer-to-peer information system based on the xor metric. *In:* Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS'02).MIT Faculty Club, Cambridge, MA.
[13] Lenzerini M. (2002). Data integration: A theoretical perspective. *In*: Proc. of the 21st ACM SIGACT SIGMOD SIGART Symp. PODS, p. 233-246.
[14] Noy Natalya Fridman, (2004). Semantic Integration: A Survey Of Ontology-Based Ap-proaches. SIGMOD Record, 33 (4) 65-70.
[15] Bernstein, P.,Giunchiglia, F., Kementsietsidis, A., Mylopoulos, J., Serafini, L., Zaihrayeu, I., (2002). Data management for peer-to-peer computing: A vision. *In*: Proc. of WebDB'02.
[16] Halevy, A., IVES, Z., Mork, P., Tatarinov, I. (2003). Piazza: Data management infra-structure for semantic web applications. *In:* Proc. of the 12th International Conference on World Wide Web, Hungary, p. 556–567.
[17] Kalfoglou Y. and Marco Schorlemmer. (2003). Ontology Mapping: the State of the Art. The Knowledge Engineering Review, 18 (1) 1-31.
[18] Haase, P., JeenBroekstra, Marc Ehrig, Maarten Menken, Peter Mika, Michal Plechawski, PawelPyszlak, BjörnSchnizler, Ronny Siebes, Steffen Staab, ChristophTempich, (2004). Bibster A Semantics-Based Bibliographic Peer-to-Peer System, *In*: Proceedings of the Third International Semantic Web Conference, p. 122-136.
[19] Nejdl, W., Wolf, B., QU, C., Decker, S., Sintek, M.,Naeve, A., Nilsson, M., Palmer, M., Risch, T. (2003). Edutella: A p2p networking infrastructure based on rdf. *In*: Proceedings of the 12th International Conference on World Wide Web.Budapest, Hungary.
[20] Crespo, A., Garcia-Molina, H. (2002). Routing indices for peer-to-peer systems. *In:* Proceedings International Conference on Distributed Computing Systems (ICDCS), p. 23-34, July.
[21] Wee Siong, Ng. Ooi, B. C., Tan, K.- L., Zhou, A. (2003). Peerdb: A p2p-based system for distributed data sharing. *In*: Intl. Conf. on Data Engineering (ICDE).
[22] Arenas, Kantere, M. V., Kementsietsidis, A., Kiringa, I., Miller, R. J., Mylopoulos, J. (2003). The Hyperion Project: From Data Integration to Data Coordination. SIGMOD Record, 32 (3) 53–38.
[23] Wadjinny, Gounbark, F., Benhlima, L., Chiadmi, L., Moujane, A. (2009). Query pro-cessing in the WASSIT mediation framework, in the seventh ACS/IEEE International Conference on Computer Systems and applications (AICCSA 2009), May 10-13, in Rabat, Morocco.
[24] El Marrakchi, M. (2009). Mise en place d'un adaptateur XQuery/SOAP pour l'interrogation des Web services à partir d'un système de médiation. M.S Thesis, Computer Sciences Department, Mohammadia Engineering School, Rabat, Morocco.
[25] Fernández, M., Malhotra, A., Marsh, J., Nagy, M., Norman, W. (2007). XQuery 1.0 and XPath 2.0 Data Model (XDM) W3C Recommendation 23, Available from http://www.w3.org/TR/xpath-datamodel/
[26] Wu, G., Li, J. Z., Feng, L. (2008). Identifying potentially important concepts and relations in an ontology, *In:* Proceedings of the 7th International Semantic Web Conference, Oct. 26-30, Karlsruhe, Germany, p. 33-49.
[27] Levy, A. Y. , Rajaraman, A. and Ordille, J. J. (1996).Querying heterogeneous information sources using source descriptions. *In*: Proceedings of the 22nd International Conference on Very Large Data Bases (VLDB'96), p. 251–262, Mumbai (Bombay), India.
[28] Lee, J. H., Kim, M. H. and Lee, Y. J. (1993). Information retrieval based on conceptual distance in is-a hierarchies. *Journal of Documentation*, 49 (2) 188–207.
[29] Wu, Z., Palmer, M. (1994). Verbs semantics and lexical selection. *In*: Proceedings of the 32nd annual meeting on Association for Computational Linguistics, p. 133–138, Morristown, NJ, USA. Association for Computational Linguistics.
[30] Leacock , C. and Chodorow, M. (1998). Combining local context and WordNet similarity for word sense identification.WordNet: An Electronic Lexical Database, 49 (2) 265–283.
[31] Vincent, D., Blondel et al. (2004). A measure of similarity between graph vertices: Applications to synonym extraction and web searching. SIAM Rev., 46 (4) 647–666.
[32] Hu, W., Jian, N., Qu, Y., Wang, Y. (2005). GMO: A Graph Matching for Ontologies. *In:* Integrating Ontologies.