



## DSD : a DaaS Service Discovery Method in P2P Environments

Riad Mokadem, Franck Morvan, Chirine Ghedira-Guegan, Djamal Benslimane

### ► To cite this version:

Riad Mokadem, Franck Morvan, Chirine Ghedira-Guegan, Djamal Benslimane. DSD : a DaaS Service Discovery Method in P2P Environments. 17th East-European Conference on Advances in Databases and Information Systems (ADBIS 2013), Sep 2013, Gênes, Italy. pp.129-137, 10.1007/978-3-319-01863-8\_15 . hal-04083796

**HAL Id: hal-04083796**

**<https://hal.science/hal-04083796>**

Submitted on 27 Apr 2023

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

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



## Open Archive TOULOUSE Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in : <http://oatao.univ-toulouse.fr/>  
Eprints ID : 12519

The contribution was presented at ADBIS 2013 :  
<http://adbis2013.disi.unige.it/>

**To cite this version** : Mokadem, Riad and Morvan, Franck and Ghedira, Chirine and Benslimane, Djamal *DSD : a DaaS Service Discovery Method in P2P Environments*. (2013) In: 17th East-European Conference on Advances in Databases and Information Systems (ADBIS 2013), 1 September 2013 - 4 September 2013 (Genoa, Italy).

Any correspondance concerning this service should be sent to the repository administrator: [staff-oatao@listes-diff.inp-toulouse.fr](mailto:staff-oatao@listes-diff.inp-toulouse.fr)

# DSD: a DaaS Service Discovery Method in P2P Environments

Riad Mokadem<sup>1</sup>, Franck Morvan<sup>1</sup>, Chirine Ghedira Guegan<sup>2</sup>, Djamel Benslimane<sup>3</sup>

<sup>1</sup>IRIT, Paul Sabatier University, 118 Rte de Narbonne, 31062, Toulouse, France

<sup>2</sup>IAE, Jean Moulin University, 6 cours Albert Thomas, 69355, Lyon, France

<sup>3</sup>LIRIS, Claude Bernard University, 69622, Villeurbanne, France

<sup>1</sup>{mokadem, morvan}@irit.fr,

<sup>2</sup>chirine.ghedira-guegan@univ-lyon3.fr, <sup>3</sup>djamal.benslimane@univ-lyon1.fr

**Abstract.** Exposing data sources through *Daas* (Data as a Service) services become increasingly important. The *DaaS* service discovery constitutes a real challenge in P2P environments. Although several data source discovery methods take into account the semantic heterogeneity problems by using several domain ontologies (DOs), most of them imposed a topology on the graph formed by DOs and mapping links. In this paper, we propose a *DaaS* Service Discovery (DSD) method without imposing any topology on this graph. Peers, using a common DO, are grouped in a Virtual Organization (VO) and connected in a Distributed Hash Table (DHT). Then, lookups within a same VO consists in a classical search in a DHT. Regarding the inter-VO discovery process, we propose an addressing system, based on the existing mapping links between DOs, to interconnect VOs. Furthermore, a lazy maintenance is adopted in order to reduce the number of messages required to update the system.

**Keywords:** Large Scale Data Distribution, Data as a Service, Data Source Discovery, Semantic Heterogeneity, Dynamicity, Performance.

## 1. Introduction

With the constant proliferation of information systems around the globe, the need for decentralized and scalable data sharing and integration mechanisms has become apparent more than ever in a wide range of applications. These applications querying heterogeneous data source spread on a huge number of peers which can join/ leave the system at any moment. Last few years saw new type of services known as *DaaS* (Data-as-a-Service) services [19] where services correspond to calls over the data sources. Besides, users' requirements increase so that their queries often need several sources, thus requiring service composition. The latter consists in combining several *DaaS* services. While initial service composition approaches have been a powerful solution for building value-added services on top of existing ones, the issues of exploiting and managing *DaaS* services in dynamic and large scale P2P environments remain an important challenge. In fact, *DaaS* services are numerous, highly heterogeneous and constantly evolving in such environments. This leads to the fact that the

services discovery process remains a more important issue in a large scale query evaluation. Indeed, the discovery process consists in searching metadata describing *DaaS* services required to process the user query. However, the main obstacle affecting the *DaaS* service discovery mechanism is the semantic heterogeneity between services, e.g., the *DaaS* service associated to the 'Doctor' concept in a medicine field is not identical to the *DaaS* service associated to the 'Doctor' concept in the university area. Indeed, resolving the semantic heterogeneity between the different concepts associated with the *DaaS* services is necessary.

The *DaaS* service discovery approaches can be classified into centralized [2] and decentralized [7]. Central registries, used to store *DaaS* services, are poor at supporting scalability in the Web context when most of the decentralized *DaaS* service discovery methods typically employ flooding and random walk to locate data which results in much network traffic. Dealing with the data source discovery related work taking into account the semantic heterogeneity problems, first works were based on the correspondence between keywords used in the data source schemas [9, 17, 18]. The major drawback of this approach is the maintenance of links in a highly dynamic environment. Other works have adopted the use of a global schema or a global ontology, employed to provide a formal conceptualization of each domain, as a pivot schema [1]. However, designing such ontology remains a complex task in front of the large number of areas in large scale environments. Later, some works proposed the using of different domain ontologies (DOs) [8, 14]. In this paper, we adopt this latter approach which is the most promising because it preserves the autonomy of each DO. Hence, each application domain is associated with one DO. Relationships links called 'mapping links' are established between these DOs in order to define correspondence links between them. In our knowledge, all methods proposed within this approach impose a particular topology on the graph formed by the DOs and mapping links. Imposing a fixed topology as in [15] is a major drawback. Indeed, there are on the Internet available DOs and mapping links between them. The topology of the graph formed by these DOs and mapping links between them is an arbitrary graph. If the topology founded is not suitable for one method, some mapping links must be defined. This is a very hard task. Hence, a good challenge consists in using the existing mapping links without imposing any topology on the graph.

In this paper, a part of the PAIRSE project<sup>1</sup>, we extend the data source discovery method proposed in [11] to support the *DaaS* service discovery with considering semantic heterogeneity between concepts associated to these services. The discovery process is particularly important that the user query response is produced from the composition/ filtering of returned *DaaS* services [4]. The proposed *DaaS* Service Discovery (DSD) method is adapted to any mappings link topology. We propose to regroup peers, by expertise domain, in a virtual organization (VO) [12]. In each VO, peers used the same DO as a pivot schema. This allows taking into account the principle of locality [10] that promotes the autonomy of each VO. For reasons of discovery process efficiency, the peers within the same VO are connected in a Distributed Hash Table (DHT) [16]. Thus, the discovery within a single VO consists in a classical lookup in a DHT. Concerning the inter-VOs *DaaS* service discovery, the translation of the sought concept between VOs is required in order to propagate the *DaaS* service discovery query. For this aim, we propose an addressing system that permits to interconnect VOs by exploiting the existing mapping links between DOs. Then, a perma-

---

<sup>1</sup> This research project is supported by the French National Research Agency under grant number ANR-SEGI-008, and available at: <https://picoforge.inria.fr/cgi-bin/twiki/view/PairseWeb/>

nent access exists from any VO to other. The proposed method takes also into account the connection/ disconnection of peers into the system (dynamicity property of P2P environments). In order to limit the excessive number of messages exchanged between peers, we adopt a lazy update of the addressing system. This permits a significant maintenance costs reduction especially in the presence of a churn effect [12]. The rest of the paper is organized as follows: Section 2 describes the *DaaS* service composition. Section 3 details the proposed DSD method. Section 4 discusses the system maintenance through our method. A simulation analysis of the proposed method is presented in section 5. The final section contains concluding remarks and future work.

## 2. DaaS Service Composition

*DaaS* services provide bridges to access data sources. Nevertheless, while individual *DaaS* services may provide interesting information alone, in a real scenario, users' queries require the invocation of several *DaaS* by adopting composition approaches.

In this work, we adopt a query rewriting based approach to compose data providing Web services proposed in [3]. Specifically, *DaaS* services are modeled as RDF parameterized Views (RPVs) over DOs. RDF (Resource Description Framework) views capture in a faithful and declarative way the semantic relationships between input and output parameters using ontological concepts and relations whose semantics are well defined in the mediated ontology. RDF views are incorporated within services description files as annotations. Users pose their queries at a given peer on a mediated ontology using SPARQL<sup>2</sup> query language. Then, the defined RDF views are exploited within WSDL files to discover locally and/or distantly services. Indeed, the peer extracts the different ontological concepts used in the query and launches service discovery requests for services annotated (via their RPVs) with these concepts, firstly in the peer where the query is posed, then the service discovery request is propagated to the others peers. The discovery process is detailed in the next section. The descriptions of discovered services are then sent back to the initial peer, where the relevant services will be selected and composed using an RDF query rewriting algorithm [4, 6]. Finally an execution plan for the composition is generated and executed to provide the user with requested data.

## 3. DaaS Service Discovery Considering Semantic Heterogeneity

P2P environments are characterized by the presence of a large number of Web services which are highly heterogeneous and constantly evolving. Throughout this section, we present the proposed *DaaS* Service Discovery (DSD) method.

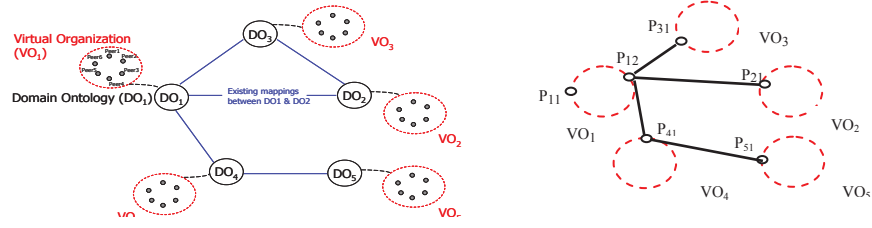
### 3.1 System Architecture and Design

Each application domain accumulates a large number of *DaaS* services. In order to preserve the autonomy of each application domain, we propose to associate each application domain with one DO [14]. Then, relationships links called 'mapping links' are established between these DOs in order to define correspondence links be-

---

<sup>2</sup> <http://www.w3.org/TR/rdf-sparql-query/>

tween them. We group peers using the same DO in a Virtual Organization (VO). This allows taking into account the principle of locality [10]. Indeed, peers having the same expertise are grouped in the same VO. Let consider a set of DOs which form an undirected graph  $G(V, E)$  with  $V$  the set of vertices presenting these DOs and  $E$  the set of edges presenting the mapping links between these DOs. We note that an edge exists between two vertices  $v_i$  and  $v_j$  in  $G$  if and only if there exists a mapping link between  $DO_i$  and  $DO_j$  presenting respectively per  $v_i$  et  $v_j$  with  $i \neq j$ .



**Fig. 1:** Example of: a Graph between DOs and Associate VOs (left), Interconnection between VOs through Access Points (right).

For each  $DO_i$  in  $G$ , we associate a virtual organization  $VO_i$ . Each VO regroups a set of peers using the same  $DO_i$  as a pivot schema for managing *DaaS services*. We affirm that  $VO_j$  is neighbor of  $VO_i$  if and only if it exist a mapping link between  $DO_i$  and  $DO_j$  used respectively per  $VO_i$  and  $VO_j$ . We notes Neighbor ( $VO_i$ ) a set of VOs, neighbors of  $VO_i$  and connected to  $VO_i$  through direct mapping links. Let also  $|\text{Neighbor}(VO_i)|$  be the number of VOs neighboring of  $VO_i$ . To ensure the completeness of discovery results, the graph  $G$  must be connected, i.e., there must be a path from any VO to another. In the rest of this paper, we suppose that  $G(V, E)$  is connected and its topology is arbitrary. Thus, there is always a path between two vertices in  $G$ . This allows a translation between two DOs, e.g., Fig.1- left shows an example of a connected graph of mapping links between  $DO_1, DO_2, \dots, DO_5$ ,  $VO_1$  regroups a set of peers using the same domain ontology  $DO_1$  as a pivot schema.

### 3.2 DaaS Service Discovery Process

Suppose that a user query  $Q$  is issued at a given peer. Suppose also that  $Q$  could not be resolved with the *DaaS services* founded at a local peer [4]. Then, its evaluation starts with the *DaaS service discovery* step. It consists to discover the metadata describing *DaaS services*, previously published and associated to the concepts present in  $Q$ . The *DaaS service registration* step consists in publishing (i) the concerned *DaaS service*, (ii) the associated RDF parameterized View (RPV) which allows to capture the semantic relationship between the *DaaS service* and the associated concepts over the corresponding DO, (iii) the set of concepts  $C$  associated to the *DaaS service*, (iv) the set of properties  $Pr$  associated with these concepts on which an additional filtering is applied to the returned *DaaS service*. Using the described configuration of VOs, *DaaS service discovery* queries can be classified into two types: (i) intra-VO *DaaS service discovery* queries and (ii) inter-VO *DaaS service discovery* queries.

**Intra-VO *DaaS service discovery* process.** The intra-VO *DaaS service discovery* process consists on the *DaaS services* research within a same VO. It does not require any concepts translation since all peers in one VO used the same ontology as a pivot schema. For this aim, we wish to (i) have an efficient mechanism for the *DaaS service*

discovery process and (ii) avoid false answers [16]. This means that if the *DaaS service* exists, we want to discover it. For efficient reasons, we have proposed to associate a structured P2P system, e.g., DHT, to each VO. DHTs have proved their efficiency with respect to the scalability and research process. In addition, they have the characteristic to avoid false answers (the case of unstructured P2P systems). The complexity to find a peer responsible of a *DaaS service* is  $O(\log(N))$  where  $N$  is the number of peers in chord [16], a DHT rooting protocol. The *DaaS service* discovery within a single VO is evaluated according to the routing system of a classic DHT [16]. However, this requires some DHT adjustments in order to index the previously published services. Indeed, we have extended the DHT catalog to support the registration of both the concerned *DaaS service* and the associated set  $\{RPV, C, Pr\}$ .

**Inter-VO *DaaS service* discovery.** An inter-VO discovery query providing from a peer  $\in VO_i$  consists to look for metadata describing a *DaaS service* available in  $VO_j$  with  $i \neq j$ . For this aim, we propose an addressing system which assures a permanent access from any VO to other in a dynamic environment in order to permit the translation of each researched concept. The proposed addressing system permits a communication between a peer  $\in VO_i$  and peers in  $neighbor(VO_i)$ . For each peer  $P_i \in VO_i$  we associate  $|Neighbor(VO_i)|$  access points. Let  $AP_i$  the access point set of  $P_i$ . Each access point  $P_j \in AP_i$  is one peer of  $VO_j \in Neighbor(VO_i)$ . Hence, when a peer  $P_i$  wants to propagate the discovery query to  $VO_i$ , access points and exiting mapping links between  $DO_i$  and  $DO_j$  are used. In order to avoid that a peer forms a bottleneck or constitutes a single point of failure, we ensure that several access points  $P_j$  of a peer  $P_i \in VO_i$  reference different peers in  $VO_j$ . Fig. 1- right illustrates examples of access points: the bold lines show mapping links between VOs, e.g.,  $P_{12} \in VO_1$  can communicate with peers of  $VO_2$  thanks to its access point  $P_{21}$ .

Every peer receiving a discovery query: (i) execute an intra-VO discovery query and (ii) propagate the discovery query towards neighbors VOs and so on. Suppose that a given peer  $P_i \in VO_i$  submits a *DaaS service* query. Hence, a lookup function is evaluated for each  $P_j \in AP_i$  in order to search the concept  $c$  in  $VO_j \in Neighbor(VO_i)$ . When  $P_i$  contacts its access point,  $c$  is translated through the existing mapping rules between DOs. To avoid an endless propagation of a discovery query, we define a Time to Live (TTL) which corresponds to the maximal path length in  $G$  than a discovery query can run, i.e., the limit of the query propagation number. The complete inter-VO discovery algorithm is described in [13]. If a *DaaS Service*, at least, is found, the response is sent to  $P_i$ . It contains: (i) metadata of the founded *DaaS services*, (ii) the translation path constituted of a sequence of edges representing the mapping links that the discovery query followed along the discovery process and (iii) the associated RPV describing the semantic relationship between each returned *DaaS service* and  $c$ . Finally, a filter is applied to each *DaaS service* through properties  $\in Pr$ .

## 4. System Maintenance

The continuous leaving/ joining of peers is very common in P2P systems. This requires the maintenance of the system. We distinguish two types of maintenance in our system: (i) the DHT maintenance and (ii) the addressing system maintenance that



impact the discovery. We will not detail the first case since the system maintenance is done by a classical maintenance of a DHT [16]. Maintaining the addressing system requires the updating of all access points, i.e., defining how the access points are updated. Recall that in P2P Chord systems [16], the connection/ disconnection of one peer generates  $\text{Log}^2(N)$  messages when  $N$  is the total number of peers in the system.

When a new peer connects to a VO, all its access points must be defined. We based on the same technique used when an access point is not available during the inter-VO discovery. The connection peer algorithm is detailed in [11]. Regarding the peer disconnection process, let's a peer  $P_{\text{Disc}} \in \text{VO}_i$  disconnects from the system. The first step is to maintain the DHT. This is a classical DHT maintenance [16]. However, the peer  $P_{\text{Disc}}$  can be an access point for a peer belonging to another  $\text{VO}_j$  (with  $i \neq j$ ). Hence, the addressing system must be updated. We have adopt a lazy maintenance as in [11]. None of the  $\text{VO}_j$  ( $i \neq j$ ) is informed by this disconnection. The access points towards this  $\text{VO}_j$  will be updated during an inter-VO *DaaS* service discovery process. Indeed, during the discovery process, the opportunity is taken to update all access points. This strategy reduces the number of messages required to update the system.

## 5. Simulation Analysis

We focus on the simulation of a data source discovery process since it is difficult to experiment these peers organized as VOs in a real platform, e.g., Grid'5000<sup>3</sup>. We based on a virtual simulated network of 10.000 homogenous peers with a single data source by peer. We also used Open Chord [16], one implementation of Chord DHT. In other hand, data sources are exposed as *DaaS* services. Thus, performances of data sources discovery process and *DaaS* service discovery process are almost equivalent.

### 5.1 Inter-VO Data Sources Discovery

We have compared performance of the DSD method to those of three other data source discovery methods taking into account the semantic heterogeneity problems: (i) data source Discovery according to the Super Peer topology method (DSP) [8], (ii) data source Discovery method in which the topology is 'Two by Two peers' (D2b2) [9] and (iii) data sources Discovery by Flooding method (DFlooding) [5].

When we deal with a single discovery query, D2b2 response times are the largest compared to other three methods. This is due to the longest path traversed to discover data sources within this method. DFlooding and DSD methods show better results in terms of response time. They have almost similar response times with a small advantage to the DFlooding method. However, the graph of mapping links between ontologies in this later must be a complete graph which requires intensive intervention of the administrator. In order to confirm the capability of the proposed DSD method to be scalable, we have varied the number of queries submitted to each peer (between 2 and 500 queries/ sec). Fig.2 shows the ratio of DSD response times over response times of the three compared solutions with a system composed of 100 VOs, i.e., with 100 peers/ VO. Experiments show that the DSD response times are significantly reduced compared to D2b2 and DSP methods. When we experiment with 10 queries/ sec, the response times generated by our method are 10 times smaller

---

<sup>3</sup> Grid'5000. [www.grid5000.org](http://www.grid5000.org)



than those generated by the DSP method. From 10 queries/ sec, the save time is more important. Compared with the DFlooding method, our response times are slowly greater (18%) when we experiment with less than 20 queries/ sec. A save time, whatever small, is obtained from 25 discovery queries/ sec, e.g., a save of 20% with 500 queries/ sec. This is due to the fact that multiple discovery queries in DSD may require the intervention of several different access points when these queries generate some bottleneck at some peers in the DFlooding method. Hence, it seems more reasonable to have more than 20 queries/ sec in a large scale environment. Furthermore, with DSD we can use DOs and mapping links available on Internet and add some mapping links if the graph founded is not connected. In the DFlooding method, a more important number of mapping links must be defined to have a complete graph.

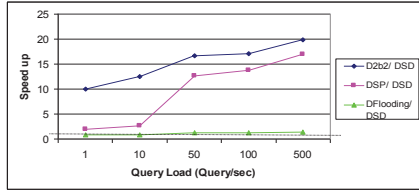


Fig. 2. Speed up (Response Times).

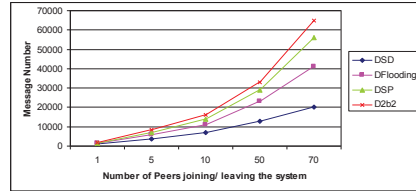


Fig. 3. Impact of the Connected/ Disconnected Peers on the System Maintenance.

## 5.2 System Maintenance: Simulation Analysis

We evaluate the maintenance costs by measuring the number of messages generated to maintain the system when peers join/leave the system but the total number of peers stays appreciatively constant (a system composed of 10 VOs with 1000 peers/ VO). It is clear that maintaining a DHT generates greatest costs especially when several peers join/leave the system. But, this is valuable for all the compared solutions. Fig. 3 shows the number of the required messages to maintain the system in the four compared solutions. The number of messages needed to maintain the system with the DSP and D2b2 methods is the higher. Indeed, if a super peer leaves or arrives in the system, all the ‘leaves’ peers should be updated by using the DSP method. The D2b2 method generates the most important maintenance cost. This is due to the topology used. When 10 peers join/ leave the system, better results are observed for the DSD method which requires less than 7200 when the DFlooding method requires 11000 messages to maintain the system. In fact, most of messages in the DSD method are essentially those required to update the DHT. The use of a lazy maintenance allows significant reduction in the number of these messages needed to update access points. Thus, access points of a peer referencing peers which have leaving the system are updated only when the discovery process occurs in the DSD method when all peers must be contacted to update their access points in the DFlooding method.

## 6. Conclusion and Future Work

The proposed *DaaS* Service Discovery (DSD) method takes into account the semantic heterogeneity problems in P2P environments. For this aim, we group all peers

using the same domain ontology (DO) in a virtual organization (VO). Within a VO, the *DaaS* service discovery process is based on a classical lookup in a DHT (intra-VO discovery). Regarding the inter-VO discovery process, we have proposed an addressing system based on the existing mappings between various DOs without imposing any topology on the graph formed by these DOs and mapping links. Our discovery method allows a permanent access between virtual organizations in a dynamic environment. Furthermore, we adopt a lazy maintenance in order to decrease the update cost generated by the continuous joining/ leaving of peers to the system.

The Simulation analysis showed a significant improvement of response times for the inter-VO discovery queries especially with an important number of simultaneous discovery queries. It shows also a significantly reduction of the maintenance cost generated by the frequent joining and leaving of peers. Further work includes more performance studies especially with a high number of peers in a real platform.

## References

1. Alking R., Hameurlain A., Morvan F.: Ontology-Based Data Source Localization in a Structured P2P Environment. Proc. of Int. symposium IDEAS, pp. 9-18, Coimbra, Portugal, (2008).
2. Atkinson C., Bostan P., Hummel O., Stoll D.: A practical Approach to Web Service Discovery and retrieval. In Proc. of Int. Conf. ICWS2007, (2007).
3. Barhamgi M., Benslimane D., Medjahed B.: A Query Rewriting Approach for Web Service Composition. IEEE Transactions on Services Computing (TSC), 3(3):206-222, (2010).
4. Barhamgi M., Ghedira C., Benslimane D., Tbahriti S. E., Mrissa M.: Optimizing DaaS Web Service Based Data Mashups. Proc. of the Int. IEEE Conf. SCC, pp. 464-471, (2011).
5. Chawathe Y., Ratnasamy S., Breslau L.: Marking Gnutella-like P2P Systems Scalable. Proc. of the Int. Conf. SIGCOMM'03, pp 407-418, Karlsruhe, Germany, (2003).
6. Chen H.: Rewriting Queries Using View for Rdf/Rdfs-Based Relational Data Integration. In Proc. of Int. Conf. on Distributed Computing and Internet Technology, ICDCIT'05, pp. 243-254, Berlin (2005).
7. Comito C., Mastroianni C., Talia D.: A Semantic-aware Information System for Multi-Domain Applications over Service Grids. IPDPS, Rome, Italy, (2009)
8. Faye D., Nachouki G. and Valduriez P.: Semantic Query Routing in SenPeer, a P2P data Management System. Proc. of the int. Conf. on network based system information systems. NBIS. Germany (2007)
9. Halevy, Z.G. Ives, P Mork and I. Tatarinov. (2003) Piazza: Data Management Infrastructure for Semantic Web Applications. In Int. Conf. WWW, Budapest, Hungary.
10. Harvey N. , Dunagan M.B. J., Jones M.B., Saroiu S., Theimer M., Wolman A.: Skipnet: A Scalable Overlay Network with Practical Locality Properties. In Proc. of USITIS, Seattle, (2003).
11. Ketata I., Mokadem R., Morvan F.: Resource Discovery Considering Semantic Properties in Data Grid Environments. Proc. of Inter. Conf. Globe'2011, pp. 61-72, France, Springer, (2011).
12. Mokadem R., Hameurlain A., Tjoa A.M.: Resource Discovery Service while Minimizing Maintenance Overhead in Hierarchical DHT Systems. Proc. Int. Conf. iWAS'10, ACM, p. 628-636, Paris, (2010).
13. Mokadem R., Morvan F., Ghedira Gueguan C., Benslimane D.: DSD: A DaaS Service Discovery in P2P Environments. IRIIT research report, RR-2013-31-FR, June 2013, (2013).
14. Navas I., Sanz I., Aldana J.F., Berlanga R.: Automatic Generation of Semantic Fields for Resource Discovery in the Semantic Web. DEXA, Bilbao, Spain, (2005).
15. Nejdl I., Wolf W., Qu B.: Edutella: AP2P networking infrastructure based on RDF. In: 11th Int. WorldWideWeb Conf., pp. 604-615, Hawaii, USA, May (2002).
16. Stoica I., Morris R., Karger D., Kaashoek M.F., Balakrishnany H.: Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. Proc. of ACM Conf. SIGCOMM, California, USA, (2001).
17. Sartiani C., Manghi P., Ghelli G., Conforti G.: XPeer: A Self-organizing XML P2P Database System. Proc. of Int. EDBT Workshop on P2P and Databases, Heraklion, Greece, (2004).
18. Siong W. Ng., Chin Ooi B., Tan K., Zhou A.: PeerDB: A P2P-based System for Distributed Data Sharing. Proc. of Int. Conf. of Data Engineering. pp. 633-644, Bangalore, (2005).
19. Vaculin R., Chen H., Neruda R., Sycara K.: Modeling and Discovery of Data Providing Services. Proc. of Int. Conf. on Web Services, pp. 54-61, Washington, USA, IEEE Computer Society, (2008).