UNIVERSITY
OF TRENTO

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

38050 Povo – Trento (Italy), Via Sommarive 14
http://www.dit.unitn.it

P2P SEMANTIC SEARCH

Uladzimir Kharkevich

May 2007

Technical Report # DIT-07-017

P2P Semantic Search

Abstract. We consider P2P Semantic Search as a process of finding documents, which are semantically, i.e., with respect to the meaning, related to the user information needs, in a document collections distributed among a group of peers, i.e., autonomous information sources. To organize documents stored on a single peer efficiently for the search, documents are classified to the user-generated classifications. Nodes in the classification specify concepts which the user is interested in. Accordingly, the whole classification specifies the user interest profile. To provide effective search in the P2P network, peers in the network should have some ways for cooperation. In our approach, related nodes in classifications on different peers are interconnected by means of semantic links, which allow peers in the network to reason about the contents of each other and efficiently cooperate. The main foci of the current PhD thesis are the development of an algorithm for P2P Semantic Search in a distributed system of interconnected classifications; the development of a P2P Semantic search system implementing the algorithm; and the development of a testing methodology allowing for a comprehensive evaluation of the system.

Keywords: P2P Search, Semantic Search, Knowledge Management.

1 Introduction

The current World Wide Web is a huge repository of documents which keeps growing significantly from year to year making it increasingly difficult to locate relevant document while searching on the web. In addition to the massiveness, the web is also a highly dynamic system. While new documents are created, existing ones are changing their content what makes the search problem even more complex.

Freedom from a centralized authority, being the one of the reasons of success of the web, is also the reason of additional challenges faced by search engines. Individuals are free to use different vocabularies while creating documents and, therefore, different words can have the same meaning (problem of synonymy) and the same word can have different meanings in different context (problem of polysemy).

The goal of a search engine in a document collection is to map a natural language query, which specifies user information needs (i.e., it denotes a set of concepts about which the user is trying to gather information), to a set of documents in the collection, which meet these needs (i.e., a set of documents which describe semantically related concepts). Nowadays search engines mainly use syntactic techniques which are based almost purely on the occurrence of words in documents and (string-based) matching of these words to the words in user queries. Consequently, they cannot deal with problems of synonymy and polysemy described above. Semantic search attempts to improve syntactic search results by using matching of concepts (not strings), taking into account relationships between these concepts. Semantic matching [19] is a key technique which is used by a semantic search.

Most major search engines are centralized systems. They attempt to create a single index for the whole Web content. But the size, dynamics, and distributed nature of the Web make it unlikely that any centralized systems can ever have complete and up-to-date knowledge about the whole network to index it. The peer-to-peer (P2P) computing paradigm appeared as an alternative to centralized search engines for searching web content. Each peer in the P2P network organizes only a small portion of the web by creating its local index and also by providing search facilities in this index. The peer receiving a query answers it locally, and propagates it to a set of the peer neighbors. Robustness and scalability are major advantages of the P2P architecture over the centralized architecture. Also, as the requirements for computational and storage resources of each peer in a P2P network are much lighter than for a server in a centralized approach, a peer's search engine can employ much more advanced techniques for search (e.g. semantic search).

The rest of this paper is organized as follows. Section 2 expands more on the problem of *P2P Semantic Search*. Section 3 describes the latest scientific achievements in the correlated areas. In section 4 the main objectives of the thesis work are defined. Section 5 concludes with the overview of work that has been done so far.

2 P2P Semantic Search

We consider *P2P Semantic Search* as a process of finding documents, which are semantically, i.e., with respect to the meaning, related to the user information needs, in a document collection distributed among a group of peers, i.e., autonomous information sources. The number of peers in the network and the number of documents stored on each peer can be very huge. It is not feasible to compute the relevance of all documents stored on each peer to the query at the query time. Therefore, we need to create some structures, which summarize essential for the search information described in documents, organize documents efficiently for a search, and provide search facilities.

To organize a local document collection of each peer in an efficient for the search manner, user-generated classifications (i.e., tree-like topic hierarchies) are used. Such classification hierarchies have always been used by humans as the most effective and intuitive way to organize their knowledge according to their subjective view of a domain of interest. Nodes in the classification specify concepts which the user is interested in. Accordingly the whole classification specifies the user interest profile. To allow automatic reasoning about classification and their contents, some formal language (such as *Propositional Description Logic* language L^C) should be used for their representation. In our approach, each classification is converted into *Normal Formal Classification* (NFC) [18]. A *Normal Formal Classification* is a rooted tree $NFC = < N, E, L^N >$ where N is a finite set of nodes, E is a set of edges on N, and L^N is a finite set of labels expressed in L^C, such that for any node $n_i \in N$, there is one and only one label $l_i^N \in L^N$ and labels of child nodes are always more specific than the labels of their parent nodes. Each document d is assigned an expression in L^C, which we call the document concept C^d and automatically classified to some node in the classification. We assume that a document d can be classified to a node n_i if and only if $C^d \sqsubseteq l_i^N$ and there is no other node n_j, such that, $C^d \sqsubseteq l_j^N$ and $l_j^N \sqsubseteq l_i^N$. The conversion of a classification into *Normal Formal Classification*, assignment of concepts to documents, and automatic document classification are described in [18].

To make peers in the P2P Network able to reason about the contents of each other, semantic links, expressed in the C-OWL [8] language, are created between related nodes in their classifications. C-OWL envisions a wide range of possible semantic relations that can hold between related nodes in different classifications (e.g., $A \xrightarrow{\equiv} B$, $A \xrightarrow{\sqsupseteq} B$, $A \xrightarrow{\sqsubseteq} B$). An example of semantically interconnected classifications is shown in Figure 1.

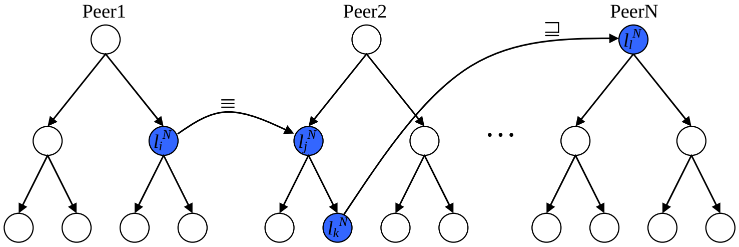


Fig. 1: Semantic Links

When the user searches for documents, he defines a query (e.g. a set of keywords), denoting a set of concepts about which the user is trying to gather information. This query is then converted into an expression in L^C using the same techniques as was used for creation of document concepts. We call this expression, a query concept C^q. The task of semantic search in the P2P network is to find documents stored in this network, whose concepts C^d are more specific than the query concept C^q.

The problem of *P2P Semantic Search* can be decomposed into three subproblems:

1. Identifying semantically relevant peers
2. Search in each relevant peer
3. Aggregation of search results from all relevant peers

3 State of the art

Search Methods in P2P networks can be classified into groups according to the following criteria:

A P2P Network structure: In *unstructured* P2P systems, peers may join and leave the network without any notification. Connections between peers are made mainly chaotically. They do not represent any predefined network structure. In *structured* P2P systems, connections between peers are fixed and data placement is related to the structure formed by peer connections. In *hybrid* P2P systems, all the peers in the network are divided into powerful peers (i.e., super peers), which are mainly responsible for network operations and weak peers connected to them.

A clustering methodology: In a *peer clustering* approach, peers which have semantically related information are organized into clusters. In *data clustering*, similar data (metadata) is placed in the same place.

Identifying semantically relevant peers: In a *blind search*, peers have no information related to the resource location. In systems which use *informed search*, peers maintain additional information about resource locations which can be useful for the search.

Markup-Scheme: In *keyword* based systems, documents have one or more predefined attributes (e.g. text, title, author, time of creation, size, etc.). In some approaches IR techniques are used to create index of local documents. In systems which use *ontology* (e.g. RDF/RDFS), metadata information is available as a set of RDF statements.

Search method: In *syntactic search*, the matching of strings and IR techniques are used. In *semantic search*, the matching of concepts (not strings) and also the relationships between those concepts are taken into account.

Table 1: Search Methods in P2P networks

	Network Structure	Clustering	Identifying semantically relevant peers	Markup-Scheme	Search method
Gnutella	Unstructured	-	Blind	Keyword	Keyword
Freenet	Unstructured	-	Informed	Keyword	Keyword
Routing Indices	Unstructured	-	Informed	Keyword	Keyword
SETS: topic-segmented overlay	Unstructured	Peers	Informed	Keyword	Keyword
Associative overlays	Unstructured	Peers	Informed	Keyword	Keyword
Interest-based overlay	Unstructured	Peers	Informed	Keyword	Keyword
ESS	Unstructured	Peers	Informed	Keyword	Keyword
”Semantic overlay network” (SON)	Unstructured	Peers	Informed	Keyword	Keyword
P2PSLN	Unstructured	Peers	Informed	XML schema	Semantic
OPDMS	Unstructured	Peers	Informed	Ontology	Semantic
Napster	Hybrid	-	Informed	Keyword	Keyword
GIA	Hybrid	-	Blind	Keyword	Keyword
FastTrack	Hybrid	-	Blind	Keyword	Keyword
EDUTELA	Hybrid	Peers	Informed	Ontology	Semantic
OAI-P2P	Hybrid	Peers	Informed	Ontology	Semantic
Bibster	Hybrid	Peers	Informed	Ontology	Semantic
RDFPeers	Structured	Data	Informed	Ontology	Keyword
CAN	Structured	Data	Informed	Keyword	Keyword
Chord	Structured	Data	Informed	Keyword	Keyword
Pastry	Structured	Data	Informed	Keyword	Keyword
Tapestry	Structured	Data	Informed	Keyword	Keyword
Mercury	Structured	Data	Informed	Keyword	Keyword
MINERVA	Structured	Data	Informed	Keyword	Keyword
pSearch	Structured	Data	Informed	Keyword	Keyword
HyperCuP	Structured	Peers	Informed	Keyword	Keyword

Gnutella [2] uses a query flooding algorithm for query routing which cannot scale well in P2P network. In Freenet [13] each peer maintains a local routing table which keeps information about neighbor peers. In [15] a peer uses Routing Indices to forward queries to neighbors that are more likely to have answers. Query topics are compared to neighbor's expertise to select relevant peers. The basic idea of [5, 14, 25, 31] is to organize peers into Similar Content Groups on top of unstructured P2P systems. Peers from the same group tend to be relevant to the same queries. A query is guided to Similar Content Group that is more likely to have answers to the given query and then the query is flooded within this group. In Semantic Overlay Networks (SONs)[16] peers that have similar documents are clustered at the same group. A predefined classification hierarchy is used to classify the peers' documents. Thus two peers belong to the same SON if some of their documents classified under the same concept in this global classification. Peers can belong to more than one SON. Thus when a node wishes to join the P2P network, it initially floods the network to obtain the classification hierarchy. It then decides which SONs to join. This can be done by classifying its documents to their associative concepts. The next step is to find nodes for each SON that it belongs to. This can be done again by flooding the network. Napster [3] adopts central servers to maintain a central directory. The rest peers register their expertise on central servers. This information is used for centralized file search over the network. Gia [12] replaces Gnutella's flooding with random walks, includes a topology adaptation algorithm and introduces a token-based flow control algorithm. FastTrack [1] is based on the Gnutella protocol and extends it with the addition of supernodes to utilize the heterogeneity between peers (computer power, bandwidth and availability) and to improve scalability.

RDFPeers [11], CAN [23], Chord [26], Pastry [17], Tapestry [30] and Mercury [7] use another approach to the routing and topology organization of P2P networks. This approach employs the idea of distributed hash tables (DHT) functionality (e.g. mapping keys onto values) on Internet-like scale. In DHT systems, documents are associated with a key which is produced by hashing, for example, the document name or the document content. The range of the output values of the hash function forms an ID space. Every peer in the system is responsible for storing a certain range of keys (or partition of ID space). The structure is formed by routing tables locally stored on individual peers. A table includes a list of other peers with addresses and range of keys they are responsible for. Such systems are highly structured. The topology is tightly controlled and documents (or information about documents) are placed at the precisely specified locations defined by their keys. Mercury [7] supports multi-attribute range queries, e.g. each query is a conjunction of ranges in one or more attributes. RDFPeers [11] is a scalable distributed RDF repository that stores each triple at three places in a multi-attribute addressable network by applying globally known hash functions to its subject, predicate, and object. In Minerva [6] and pSearch [27] DHT holds only compact, aggregated meta-information about the peers' local indexes which is used to efficiently select promising peers from across the peer population that can best locally execute a query. HyperCuP [24] proposes a graph topology which allows for very efficient broadcast and search which intend to reach all peers in the network with the minimum possible number of message.

All systems discussed so far support only keyword-based local search on the peer. Semantic search on a peer is used in [4, 9, 22, 28, 32]. A semantic link peer-to-peer network (P2PSLN) [32] specifies and manages semantic relationships between peers' data schemas. A semantic-based peer similarity measurement is used for efficient query routing. A schema mapping algorithm is used for query reformulation and heterogeneous data integration. Ontology-based P2P data management system (OPDMS) [28] is based on ontology mapping and query processing. Edutella [22], OAI-P2P [4] and, Bibster [9] are built on JXTA framework and aim to combine metadata with P2P networks. Each peer is described and published using an advertisement, which is an XML document describing a network resource. For example in the Bibster [9] system, these expertise descriptions contain a set of topics that the peer is an expert in. Peers use a shared ontology to advertise their expertise in the Peer-to-Peer network.

In this section we didn't discuss database-based P2P information systems. The interested reader is referred to [29] for a detailed discussion of such systems.

4 Objectives of the thesis work

The primary objectives of the thesis work are the development of the algorithm for P2P Semantic Search; the development of the P2P semantic search system implementing the algorithm; and the development of the testing methodology allowing for a comprehensive evaluation of the systems. These are not trivial tasks to perform, and they involve a number of different issues to be solved. In particular, the thesis work includes:

1. *Development of an algorithm for Semantic Search in a single classification.*
2. *Integration of "Syntactic Search" and "Semantic Search" techniques in an efficient manner.*
3. *Development of an algorithm for P2P Semantic Search.*
4. *Quantifying the quality of the query answers of P2P Semantic Search.*
5. *Development of the logical and physical architecture of the P2P Semantic Search system implementing the algorithm.*
6. *Design and implementation of a P2P Semantic Search system based on the developed architecture.*
7. *Performance study for the developed P2P Semantic Search system.*

5 What has been done so far

As a first step toward the development of the P2P Semantic Search algorithm the algorithm of Semantic Search in a single classification was developed.

We defined the answer A^q to a query q as the set of documents, whose concepts $C^d \in L^C$ are more specific than the query concept $C^q \in L^C$.

$$A^q = \{d \mid C^d \sqsubseteq C^q\}. \tag{1}$$

After NFC was created and all documents from the document collection were classified in it the searching problem can be divided into three sub-problems:

1. Search for relevant nodes n_i (i.e., nodes which can contain relevant documents)
2. Search for a set of relevant documents A_i^q in each relevant node n_i.
3. Aggregation of search results A_i^q from all relevant nodes into final result A^q.

5.1 Search for the relevant nodes:

For any query q we can divide a set of nodes N in the classification into three disjoint subsets:

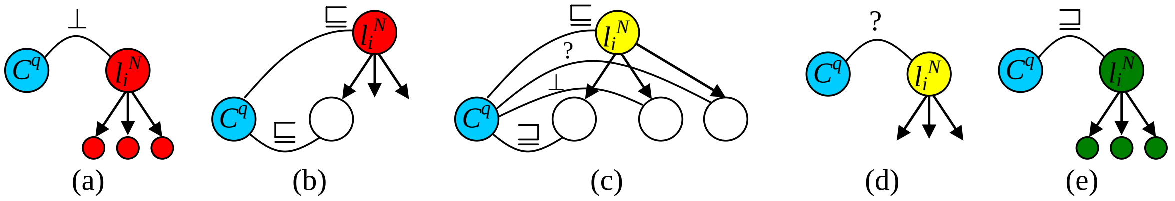
N_n^q - there is no documents classified to the nodes $n_i \in N_n^q$ which are answers to the query
N_a^q - there is no documents classified to $n_i \in N_a^q$ which are not answers to the query
N_s^q - there could be documents classified to $n_i \in N_s^q$ which are answers to the query.

In order to find relevant nodes (i.e., $n_i \in (N_s^q \cup N_a^q)$) we recursively check all nodes in the classification starting from the root node. Semantic Matching technique [19] is used to produce semantic relations between concept at query and concept at node which is checked. Five possible situation which we can find when we check node n_i are depicted on Figure 2.

5.2 Search in a single node

The task is to find relevant to the query documents from all documents classified to the node. If concept at node is more specific than concept at query (i.e., $n_i \in N_a^q$) than a query answer is a complete set of documents classified to this node. If concept at node is not more specific than concept at query and these concepts are not disjoint (i.e., $n_i \in N_s^q$) a query answer is a set of documents classified to this node which satisfy formula 1: $A_i^q = \{d \in n_i \mid C^d \sqsubseteq C^q\}$.

We compute a query answer A_i^q in a node $n_i \in N_s^q$ by sequentially checking each document classified to this node. Node in the classification can potentially contains a big number of documents



a) $(C^q \sqcap l_i^N) \sqsubseteq \bot \Rightarrow$ neither node n_i nor any of its children can have relevant documents $(n_i \in N_n^q)$
b) $C^q \sqsubseteq l_i^N$ and $\exists n_j$ such that $C^q \sqsubseteq l_j^N, l_j^N \sqsubseteq l_i^N \Rightarrow$ node n_i cannot have relevant documents $(n_i \in N_n^q)$
c) $C^q \sqsubseteq l_i^N$ and $\nexists n_j$ such that $C^q \sqsubseteq l_j^N, l_j^N \sqsubseteq l_i^N \Rightarrow$ node n_i can have relevant documents $(n_i \in N_s^q)$
d) $C^q \ ?\ l_i^N \Rightarrow$ node n_i can have relevant documents $(n_i \in N_s^q)$
e) $C^q \sqsupseteq l_i^N \Rightarrow$ node n_i and all its children have only relevant documents $(n_i \in N_a^q)$

Fig. 2: Relevant nodes

classified to it. Checking each document can be not feasible at the run time. To provide efficient search in a single node, we can create a new structure which organizes documents classified to this node. As an example of such structure can be a DAG where nodes are concepts at documents C^d and arcs are " $\sqsupseteq$ " relations. This DAG can in general be constructed in many ways. To make management of documents more efficient, we add two requirements:

1. Minimality (i.e., there is no arcs you can delete without losing information)
2. Completeness (i.e., there is no arcs you can add which increase information)

5.3 Aggregation of results

We compute a search result in a classification A^q by taking the union of search results A_i^q from all relevant nodes.

$$A^q = \bigcup_i A_i^q. \tag{2}$$

The task is to order a set of documents in A^q according to their relevance to the query. To do it, we need either to estimate the relevance $R(d, q)$ of each single document d to the query q by some function $R(C^d, C^q)$, or to be able to compare any two documents and decide which of them is more relevant to the query. To compute semantic relevance $R(C^d, C^q)$, we propose to use measure of *Semantic Similarity* $SS(C^d, C^q)$ which indicates how much information concept at document C^d and concept at query C^q share in common.

We define *Semantic Similarity* between any two concepts C_1 and C_2 as follows:

$$SS(C_1, C_2, \mathcal{I}) = \frac{size((C_1 \sqcap C_2)^{\mathcal{I}})}{size((C_1 \sqcup C_2)^{\mathcal{I}})}, \tag{3}$$

where
$\mathcal{I} = (\Delta^{\mathcal{I}}, (.)^{\mathcal{I}})$ - the *interpretation*
$\Delta^{\mathcal{I}}$ - the domain of the interpretation (non-empty set of elements)
$(.)^{\mathcal{I}}$ - interpretation function, which assigns to every concept C a set $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$.
$size((.)^{\mathcal{I}})$ is a number of elements in $(.)^{\mathcal{I}}$

As an example of the interpretation domain we can use a set of all concepts in the whole document collection. Interpretation function in this case assigns to every given concept C a set of all concepts which are more specific then C, or which are equivalent to C.

6 Future work

As follows from Section 2, three main questions to be answered while designing P2P Semantic Search engine are how to identify semantically relevant peers in the network where mainly all peers know only about a small subset of other peers present on the network; how to search in each relevant peer using semantics and/or syntax of the query; and how to aggregate search results from a set of heterogeneous peers, which can have relevant documents, into a global search result, that will be shown to the user.

The efficiency of a search technique in a P2P Network is essentially determined by the way how peers are organized in the overlay network and which query propagation technique is used in this network. To prevent network overload, users query should be propagated only to a set of semantically relevant peers or at least the number of irrelevant peers which are queried should be minimized. As described in Section 2 we are working in the network, where peers are interconnected by the means of semantic links created between semantically related nodes in their classifications. Such links stored on a peer allow this peer to reason about the contents of other peers connected by these links. It is necessary to find efficient way of exploiting such links which will give peer ability to reason about the content of the whole network.

The task of semantic search is not to supersede syntactic search but to improve it by replacing keywords by concepts and taking into account concepts interdependencies. So it is important to study the optimal balance between syntactic and semantic methods while solving a search problem. Some existing search techniques already use semantic methods to augment keyword search. Most of them make use of some thesaurus ontology (e.g. WordNet). For example in [10] WordNet is used for query expansion and in [21] WordNet information is used for selecting particular meaning of query terms during query formulation.

Peer receiving query results from a set of relevant peers should merge these results into a single list of ranked documents. So a measure of relevance to the query should be computed for each document. Important parameters which should be taken into account while scoring a particular document are: the data quality characteristics assigned to the document by the peer providing it; the behavioral characteristics of the peer providing the document; and the quality of links used during query/answer propagation in the retrieving of the considered document. In aggregate, a notion of good enough answer, introduced in [20], can be exploited in solving the problem of global result aggregation.

References

1. Fasttrack: http://www.fasttrack.nu/.
2. Gnutella: http://www.gnutella.com/.
3. Napster: http://www.napster.com/.
4. B. Ahlborn, W. Nejdl, and W. Siberski. Oai-p2p: A peer-to-peer network for open archives. In *International Conference on Parallel Processing Workshops (ICPPW02)*, 2002.
5. M. Bawa, G. Manku, and P. Raghavan. Sets: Search enhanced by topic segmentation. In *Proceedings of The 26th Annual International ACM SIGIR Conference*, pages 306–313, 2003.
6. Matthias Bender, Sebastian Michel, Peter Triantafillou, Gerhard Weikum, and Christian Zimmer. Minerva: Collaborative p2p search. In *Proceedings of VLDB*, pages 1263–1266, 2005.
7. A. Bharambe, M. Agrawal, and S. Seshan. Mercury: Supporting multi-attribute range queries. In *Proceedings of ACM SIGCOMM*, 2004.
8. P. Bouquet, F. Giunchiglia, F. van Harmelen, L. Serafini, and H. Stuckenschmidt. Contextualizing ontologies. In *Journal of Web Semantics*, volume 1, pages 325–343, 2004.
9. J. Broekstra, Marc Ehrig, Peter Haase, Frank van Harmelen, Maarten Menken, Peter Mika, Bjorn Schnizler, and Ronny Siebes. Bibster - a semantics-based bibliographic peer-to-peer system. In *The Second Workshop on Semantics in Peer-to-Peer and Grid Computing (SEMPGRID'04)*, New York, May 2004.
10. D. Buscaldi, P. Rosso, and E. S. Arnal. A wordnet-based query expansion method for geographical information retrieval. In *Working Notes for the CLEF Workshop*, 2005.
11. M. Cai and M. Frank. Rdfpeers: A scalable distributed rdf repository based on a structured peer-to-peer network. In *The 13th International World Wide Web Conference (WWW)*, pages 17–22, 2004.
12. Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and L. Breslau. Making gnutella-like p2p systems scalable. In *Proceedings of SIGCOMM*, 2003.

13. Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong. Freenet: A distributed anonymous information storage and retrieval system. *Lecture Notes in Computer Science*, 2009:46–66, 2001.
14. E. Cohen, H. Kaplan, and A. Fiat. Associative search in peer to peer networks: Harnessing latent semantics. In *Proceedings of IEEE INFOCOM*, volume 2, pages 1261–1271, 2003.
15. A. Crespo and H. Garcia Molina. Routing indices for peer-to-peer systems. In *Proceedings of the 22nd IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2002.
16. Arturo Crespo and Hector Garcia-Molina. Semantic overlay networks for p2p systems. Technical report, Stanford University, 2002.
17. P. Druschel and A. Rowstron. Pastry: scalable, distributed object location and routing for large-scale peer-to-peer systems. In *Proceedings of ACM SIGCOM*, 2001.
18. F. Giunchiglia, M.Marchese, and I. Zaihrayeu. Encoding classifications into lightweight ontologies. In *Proceedings of ESWC'06*, 2006.
19. F. Giunchiglia, M. Yatskevich, and E. Giunchiglia. Efficient semantic matching. In *Proceedings of ESWC'05*, 2005.
20. F. Giunchiglia and I. Zaihrayeu. Making peer databases interact - a vision for an architecture supporting data coordination. In *Proceedings of the International workshop on Cooperative Information Agents (CIA)*, pages 18–35, 2002.
21. P.M. Kruse, A. Naujoks, and D. Roesner M. Kunze. Clever search: A wordnet based wrapper for internet search engines. In *Proceedings of the 2nd GermaNet Workshop*, 2005.
22. W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmer, and T. Risch. Edutella: A p2p networking infrastructure based on rdf. In *Proceedings to the Eleventh International World Wide Web Conference*, 2002.
23. S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In *Proceedings of SIGCOMM*, 2001.
24. M. Schlosser, M. Sintek, S. Decker, and W. Nejdl. Hypercup - hypercubes, ontologies and efficient search on p2p networks. In *Workshop on Agents and P2P Computing*, 2002.
25. Kunwadee Spripanidkulchai, Bruce Maggs, and Hui Zhang. Efficient content location using interest-based locality in peer-to-peer systems. In *Proceedings of IEEE INFOCOM*, volume 3, pages 2166–2176, 2003.
26. Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of SIGCOMM*, 2001.
27. Chunqiang Tang, Zhichen Xu, and Sandhya Dwarkadas. Peer-to-peer information retrieval using self-organizing semantic overlaynetworks. In *Proceedings of ACM SIGCOMM*, pages 175–186, 2003.
28. H. Xiao and I. F. Cruz. Ontology-based query rewriting in peer-to-peer networks. In *Proceedings of the 2nd International Conference on Knowledge Engineering and Decision Support*, pages 11–18, 2006.
29. I. Zaihrayeu. *Towards Peer-to-Peer Information Management Systems*. PhD thesis, University of Trento (Italy), 2006.
30. B. Y. Zhao, J. Kubiatowicz, and A. Joseph. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Technical report, Computer Science Department, University of California, 2001.
31. Yingwu Zhu and Yiming Hu. Ess: Efficient semantic search on gnutella-like p2p system. Technical report, Department of ECECS, University of Cincinnati, 2004.
32. Hai Zhuge, Jie Liu, Liang Feng, Xiaoping Sun, and Chao He. Query routing in a peer-to-peer semantic link network. *Computational Intelligence*, 21(2):197–216, 2005.