Querying Unstructured and Structured Peer-to-Peer Networks: Models, Issues, Algorithms

Alfredo Cuzzocrea
ICAR Institute & DEIS Department
University of Calabria
Italy
cuzzocrea@si.deis.unical.it

Journal of Digital
Information Management

ABSTRACT: *Peer-to-Peer (P2P) networks are defined as a collection of peers that expose client/server functionalities simultaneously. P2P systems, built on top of P2P networks, support information sharing primitives and lookup mechanisms of data objects located on peers. It has been demonstrated that the P2P paradigm is able to efficiently capture models and, above all, dynamics of modern applications, beyond limitations of conventional produce/consumer paradigms. Traditionally, P2P primitives and mechanisms have been implemented by means of keyword-based search and matching operations. Modern P2P systems require more complex query functionalities, beyond capabilities of DBMS-inspired models and algorithms. As a consequence, the issue of efficiently querying the underling P2P network is gaining momentum in the research community. Indeed, querying P2P networks, which can be in the unstructured and structure modes, plays a critical role in next-generation P2P systems, as its performance heavily affect the efficiency of the overall system. Starting from these considerations, in this paper we propose a survey on models, issues and algorithms for querying unstructured and structured P2P systems. We also put in evidence similarities and differences of state-of-the-art proposals appearing in literature, with critical discussion, and we provide a rigorous taxonomy of P2P query strategies. Finally, we complete our analytical contribution via highlighting future directions in this research field.*

Categories and Subject Descriptors
C.2.2 [Network protocols]; Routing protocols: **C.2.4[Distributed Systems**]; **H.2.4 [Systems**]: Query processing

General Terms
Peer-to-peer networks, Querying P2P networks

Keywords: Querying P2P networks; Query models, Issues and algorithms for P2P networks; Taxonomies of P2P query strategies, Routing P2P networks

Received 22 December 2006; Revised 12 March 2007 and 29 April 2007; Accepted 05 May 2007

1. Introduction

Since the last decade, there is a growing interest for *Peer-to-Peer* (P2P) *Systems*, mainly because they fit a wide number of real-life applications. Digital libraries are only a significant instance of P2P systems, but it is very easy to foresee how large the impact of P2P systems on innovative and emerging scenarios, such as *e*-government and *e*-procurement, will be during next years.

P2P networks, the underlying infrastructure for P2P systems, are natively built on top of a very large repository of data objects (e.g., files), which is intrinsically distributed, fragmented, and partitioned among *participant peers*. P2P users are usually involved in (*i*) retrieving data objects containing information of interest, like video and audio files, and (*ii*) sharing information with other (participant) users/peers. From the *Information Retrieval* (IR) perspective, P2P users (*i*) typically submit short, loose queries by means of keywords derived from natural language-style questions (e.g., "*find all the music files containing Mozart's compositions*" is posed through the keywords "*composition*" and "*Mozart*"), and (*ii*), due to resource-sharing purposes, are usually interested in retrieving as result a *set* of data objects rather than only one. Based on such set of items, well-founded IR methodologies like *ranking* can be successfully applied to improve system query capabilities, thus achieving performance better than that of more traditional database-like query schemes. Furthermore, the above-described mechanism is "self-alimenting" as intermediate results can be then re-used to share new information, or to set and specialize new search/query activities. In other words, from the database perspective, P2P users typically adopt a semi-structured (data) model for querying data objects rather than a structured (data) model. On the other hand, efficiently accessing data in P2P systems, which is an aspect directly related with the issues above, is a relevant and still un-completely solved open research challenge [1].

Traditional functionalities of first-generation P2P systems are currently being extended by adding to their native capabilities (i.e., file sharing primitives and simple lookup mechanisms based on partial- or exact-match of search strings) useful (and more complex) *Knowledge Representation and Extraction Techniques*. Achieving the definition of new knowledge delivery paradigms over P2P networks is the underlying goal of this effort; in fact, the completely decentralized nature of P2P networks, which enable peers and data objects to come and go at will, allows us to (*i*) successfully exploit self-alimenting mechanisms of knowledge production, and (*ii*) take advantages from innovative knowledge representation and extraction models based on semantics, metadata management, probability etc. All considering, we can claim that, presently, there is a strength, effective demand for enriching P2P systems with functionalities that (*i*) are proper of *Information Systems* (IS), such as *Knowledge Discovery* (KD)- and IR-style data object querying, and (*ii*) cannot be supported by the actual data representation and query models of traditional P2P systems. More specifically, knowledge representation and management techniques mainly concern with the modeling of P2P systems, whereas knowledge discovery techniques (implemented via IR functionalities) mainly concern with the querying (i.e., knowledge extraction) of P2P systems.

Following this trend, a plethora of P2P query techniques have been proposed recently, each of them focused on covering a particular/specific aspect of the knowledge extraction phase. Indeed, query strategy used to retrieve information and knowledge is the most relevant characteristic of any P2P system, manly from the database research perspective.

According to these considerations, in this paper we (*i*) provide a taxonomy of P2P query strategies for both unstructured and structured networks, and (*ii*) put in evidence similarities and differences among the investigated techniques. This taxonomy helps us to keep track of the large number of proposals that have come up in the last years, and to support future research in this leading area.

1.1. Paper Outline

The remaining part of the paper is organized as follows. In Section 2, we provide a brief overview on unstructured and structured P2P networks, which are both the focus of our work. In Section 3, we provide a meaningful example of P2P system, and highlight challenges and open issues of querying such systems. In Section 4, we provide a comprehensive survey on models, issues and algorithms related to querying unstructured and structured P2P networks, along with a rigorous taxonomy of P2P query strategies. In Section 5, we propose a comparative analysis of the investigated strategies, by also highlighting benefits and limitations of techniques founding on them. Finally, in Section 6, we derive conclusions and list future directions in this research field.

2. Unstructured and Structured P2P Networks

First experiences of P2P systems, such as *Gnutella* [18], *KaZaA* [26], and *Napster* [35], mainly focused on data management issues on P2P networks, have been oriented towards designing techniques for which *sharing* data objects, and generating large *communities* of participant peers are the most relevant goals. Under this assumption, two reference architectures have gained a leading role for P2P systems, each of them addressing two different ways of retrieving data objects by querying: *unstructured P2P systems* and *structured P2P systems*.

As regards unstructured P2P systems, there are three main variants. In the first one (e.g., Napster [35]), a centralized index storing a directory of all data objects currently available on the P2P network is located in a certain peer, whose identity is known to *all* the peers. When a participant peer p_i receives a request for a missing data object, it (*i*) performs a query against the peer containing the centralized index in to retrieve the name of the (participant) peer p_j where the required data object is stored, and (*ii*) re-directs the request towards p_j. In the second variant (e.g., Gnutella [18]), there not exists any centralized index, as the latter can be source of failures, and each participant peer needs to maintain only (*i*) information about its own data for supporting data object lookups, and (*ii*) information about its neighboring peers for routing requests coming from other peers; given such a scheme, a request for a missing data object is flooded from a peer p_i towards other peers via the neighboring peers of p_i. In the last variant (e.g., KaZaA [KaZaA, WWW]), peers connect to a *super-peer* that builds an index over the data objects shared by its set of peers. In addition to this, each super-peer keeps information about neighboring super-peers in the system, and queries are routed among super-peers. Scalability is the most important drawback for unstructured P2P systems: in fact, when the number of participant peers grows, the described query mechanism can become very inefficient, as flooding the P2P network for each data object request can became (very) resource-intensive.

In structured P2P systems, all the available data objects are indexed by a high performance indexing data structure (such as *Distributed Hash Tables* (DHTs)) that is distributed among participant peers. Some examples of P2P systems adhering to such an architecture are: *Chord* [45], *Content-Addressable Network* (CAN) [38], *Tapestry* [3], and *XPath for P2P* (XP2P) [2; 3]. Compared with the previous class of P2P systems, structured P2P systems introduce the important advantage that, at query time, they do not flood the requests across the P2P network, but, indeed, the required data objects are quickly localized thanks to the distributed index. As a consequence, structured P2P systems ensure high performance. Nevertheless, an important limitation is represented by the need of updating the distributed index, particularly when highly-dynamic P2P networks are handled, as dynamics of peers quickly modify the network topology with a very high *churn rate*, which refers to the tendency of peers to join and leave the network [21]. In other words, structured P2P networks suffer of scalability issues.

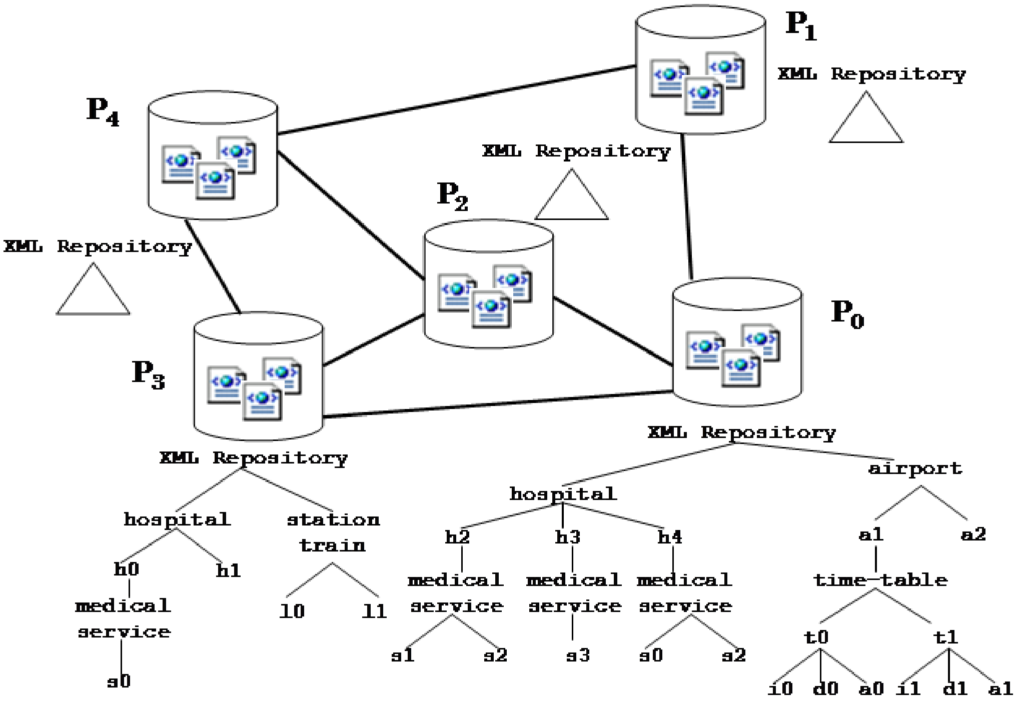

Figure 1. A healthcare P2P system built on top of distributed XML repositories

3. Querying P2P Networks: Challenges and Open Issues

Figure 1 shows an example P2P system where (peer) data repositories are stored in forms of XML files, what is a common solution appeared recently. Specifically, we are referring a P2P system focused to the healthcare context. In such system, peers store information about hospitals and medical services in a certain urban area, and other related and useful information such as locations of airports and train stations, time-tables of public services, hotels and availabilities etc concerning with that area. Users of such system are interested in retrieving (useful) knowledge by means of XQuery-formatted queries over distributed XML repositories.

As related to the problem of querying a P2P system like that depicted in Figure 1, several challenges and open issues can be identified. Among them, we list the following ones.

- **Schema-aware vs. schema-less query methods.** In a P2P network, data schemas may be available or not. This poses different requirements and problems concerning querying P2P data. For instance, if schemas are available, then queries can be posed accordingly, thus reducing many NULL values and false positives. If schemas are not available, then queries must be posed via keywords mainly, so that different query strategies capable of dealing with the lack of a fixed schema need to be devised.

- **Integration issues in the presence of schemas.** If schemas are available, integration issues arise accordingly, as the strongly distributed and decentralized nature of P2P networks can cause the phenomenon of having peers that store similar information modeled according to even-very-different schemas. This is a realistic bottleneck to be considered, as it can be source of performance decrease.

- **Network flooding.** Network flooding refers to the phenomenon of moving from a peer to another in search for specific data objects. This phenomenon can easily become unbounded in large and highly-dynamic P2P networks, thus originating a great number of hops and, as a consequence, introducing excessive computational overheads that limit the effectiveness of the overall search task. Usually, query techniques for P2P networks aim at avoiding network flooding, and bounding the number of hops allowed during the search task.

- **Understanding the semantics of neighborhood.** Although very often neglected, the concept of neighborhood is critical for querying P2P systems, as it can affect query performance significantly. Usually, neighboring peers of a given peer p_i are defined as those peers connected to p_i directly, or whose distance from p_i is within a given threshold E. This "algorithmic" vision needs to be revised. As an example, exploiting semantics in order to model the *conceptual distance* among peers is a promising direction of research (e.g., [10]). Also, this approach allows us to move from "traditional" queries, i.e. queries looking for information content stored in peers, to *conceptual queries*, i.e. queries looking for concepts stored in peers (and related information content). Finally, semantics allows us to definitively devise intelligent techniques for routing queries across peers, beyond actual capabilities of DBMS-inspired solutions.

- **Query classes.** Typical queries of P2P networks are very simple, and mainly focused to keyword-based search. Indeed, following innovative requirements posed by novel and emerging applications, there is a tough need for enriching the class of queries supported by P2P systems. To give examples, *range-queries*, *k-NN queries* and *top-k queries* are relevant classes of queries that, if embedded in P2P middleware, would allow us to improve the knowledge processing capabilities of P2P systems relaying on top of them.

- **Join queries.** P2P users typically submit queries involving multiple peer data repositories. This because information is very often distributed, whereas users wish to access information in a centralized manner, in order to extract summarized knowledge, e.g. useful for decision making. The mechanism above is implemented-in-practice in terms of *join queries* over distributed peer data repositories. Join queries add to the actual result all data objects located in distributed peers such that these objects jointly satisfy a given set of predicates. Contrarily to classical join queries in distributed databases, in the P2P context these queries introduce novel and unrecognized challenges, such as the issue of defining the non-blocking semantics in the presence of missing data objects extracted from remote peers.

- **Dealing with imprecise/incomplete information.** Peers may store imprecise/incomplete information. This because of the even-complex processes according to which knowledge is produced, processed and delivered across peers. The presence of such kind of information imposes us to consider innovative query models and algorithms able to provide *consistent* answers to such *inconsistent* information sources. Logic-based approaches are promising directions of research with respect to this specific goal.

Due to space limitations, it is not possible here to include all challenges and open issues of interest for us. However, the above-listed points can be considered, without loss of generality, as the most relevant research challenges and open issues of querying P2P networks.

4. A Taxonomy of P2P Query Strategies

It is well established (e.g., [47; 51]) that P2P systems are mainly characterized by their proper query strategies allowing us to retrieve useful knowledge in form of data objects (e.g., files). According to this vision, in this Section, we propose a taxonomy of state-of-the-art P2P query strategies (summarized results are shown in Table 1).

The first general classification distinguishes between *Keyword-based P2P* (KbP2PS) and *Object Identifier-based P2P* (OIDbP2PS) *Systems*, by looking at the atomic construct they use to drive the search mechanism. In KbP2PS, traditional keywords are used to drive the search through peers, whereas, in OIDbP2PS, object identifiers are implemented on peers to enhance query performance by biasing the search towards specific sub-sets of peers (in particular, due to the decentralized nature of P2P systems, object identifiers are usually embedded into distributed indexing data structures such as DHTs). It should be noted that both KbP2PS and OIDbP2PS can be implemented on top of either unstructured or structured P2P systems, meaning that the search mechanism is independent by the specific system topology, even if different performance are achieved. Secondly, state-of-the-art proposals are classified according to their query strategies that concern with how to route (through peers) messages needed to answer queries, thus retrieving information and knowledge.

4.1. Basic Search Techniques

Among the *Basic Search Techniques* (BST), the *Breadth First Search* (BFS) is one of the most popular ways of supporting query evaluation over P2P networks. In BFS, a peer p_i receiving a query message q from a sender peer p_j first forwards q to all its neighboring peers, other than p_j, and then searches its local repository for relevant matches. Furthermore, if a peer p_k reached by q finds a match in its repository, it sends across the network the "hit" message along with (*i*) the identifiers needed to download from it the data objects of interest, and (*ii*) the state of its network connectivity. Finally, if p_j receives hit messages from more than one peer, it may decide to download the retrieved documents from peers on the basis of their network connectivity states. BFS, which has been one of the first query strategies implemented in P2P networks, such as in Gnutella [18], KaZaA [26] and Napster [35], presents the advantage of being very simple so that no excessive computational overheads are introduced in the middleware software. Contrarily to this, BFS performance is usually very poor due to an inefficient network resources utilization that generates a lot of service messages across the P2P network, and peers with low bandwidth can become serious bottlenecks for such search mechanism. However, equipping query messages with the *Time-To-Live* (TTL) parameter, which determines the maximum number of hops allowed for any query message, can limit network flooding and sensitively increase performance.

4.2 Random Search Techniques

Random Search Techniques (RST) represent a simple-yet-effective derivation from BST. Kalogeraki *et al.* [Kalogeraki et *al.* [25] propose a significant extension to the "naïve" version of BFS, called *Random Breadth First Search* (RBFS), which consists in propagating the query message from a peer to a randomly determined sub-set of its neighboring peers rather than all of them. A setting parameter establishes how many neighboring peers must be involved by the propagation of the query message (e.g., if the parameter is equal to 0.5, then the query message is propagated to half of all the neighboring peers, chosen at random). The major benefit of the RBFS approach consists in the fact that performance of the BFS approach is dramatically improved yet ensuring low computational overheads because of the random choice does not require any global knowledge. On the other hand, being RBFS a probabilistic technique, it could happen that large segments of the P2P network are neglected by the random choice, thus reducing the efficiency of the query task. Lv *et al.* [31] present the *Random Walkers Algorithm* (RWA), according to which each peer forwards the query message (called, in this context, *walker*) to another of its neighboring peers at random. To improve performance and reduce query time, the original idea of using one walker only is extended to the usage of $k > 1$ walkers, which are consecutively sent from the sender peer. RWA resembles RBFS but, indeed, in RBFS the query message is propagated to a sub-set of peers instead that at only one for time (as in RWA); as a consequence, in RBFS the number of service messages across the P2P network can become exponential, whereas in RWA such a number is bounded by a linear complexity [31]. The *Adaptive Probabilistic Search* (APS), proposed by Tsoumakos and Roussopoulos [48], is inspired from RWA with the difference that in APS each peer p_i implements a local data structure that captures the relative probability of each neighboring peer p_j to be chosen as the next hop for future requests. Furthermore, while RWA forwards the

Class	***Search Kind***	***P2P Query Technique***
BST	KbP2PS	[Gnutella, WWW (18)], [KaZaA, WWW (26)], [Napster, WWW]
RST	KbP2PS	[Kalogeraki et al., 2002 (24)], [Lv et al., 2002 (31)], [Tsoumakos and Roussopoulos, 2003b (48)]
IST	KbP2PS	[Gen Yee and Frieder, 2005 (16)], [Meng et al., 2002 (33)], [Sripanidkulchai et al., 2003 (44)], Zeinalipour-Yazti et al., 2005 (52)]
SST	KbP2PS	[Galanis et al., 2003 (14)], [Gong et al., 2005 (18)], [Koloniari and Pitoura, 2004 (28)]; [Yang and Garcia-Molina, 2002 (50)]
IndST	OIDbP2PS	[Aberer, 2001 (1)], [Bonifati and Cuzzocrea, 2006 (2)], [Bonifati et al., 2004 (3)], [Bremer and Gertz, 2003 (4)], [Clarke et al., 2000 (7)], [Crainiceanu et al., 2004 (8)], [Crespo and Garcia-Molina, 2002 (9)], [Gibbons et al., 2003 (17)], [Gupta et al., 2003 (22)], [Rhea et al., 2001 (39)], [Loo et al., 2004] (30), [Morpheus, WWW], [Ratnasamy et al., 2001 (38)], [Sartiani et al., 2004 (42)], [Stoica et al., 2001 (45)], [Zhao et al., 2001 (54)]
DIRT	KbP2PS	[Callan, 2000 (6)], [Gauch et al., 1999 (15)], [Ogilvie and Callan, 2001 (37)], [Xu and Callan, 1998 (49)] SemST KbP2PS [Cai and Frank, 2004 (5)], [Crespo and Garcia-Molina, 2003 (6)], [Cuzzocrea, 2005 (11)], [Cuzzocrea, 2006 (12)], [Deerwester et al., 1999 (13)], [Golub and Loan, 1996 (19)], [Halaschek et al., 2004 (23)], [Halevy et al., 2003 (24)], [Kokkinidis and Christophides, 2004 (27)], [Li et al., 2003 (29)], [Nejdl et al., 2003 (36)], [Salton and Buckley, 1990 (40)], [Salton et al., 1975 (41)], [Tang et al., 2003 (46)], [Zhang et al., 2001 (53)], [Zhu et al., 2006 (55)]

Table 1. A taxonomy of P2P query strategies

walkers at random, APS exploits knowledge derived from previous searches to model the behavior of walkers on a probabilistic base. In [48], experimental results presented by authors show that APS outperforms RWA.

4.3 Intelligent Search Techniques

Beyond the previous "basic" approaches, another line of research aims at integrating intelligent techniques, perhaps inherited from similar experiences in related-but-different scientific disciplines, into the P2P middleware as to enforce the quality of the search task. We name such a class of proposals as *Intelligent Search Techniques* (IST). The *Intelligent Search Mechanism* (ISM), proposed by Zeinalipour-Yazti *et al.* [52], belongs to the latter technique class, and represents a novel approach for supporting query functionalities over P2P networks by (*i*) minimizing the number of messages sent among the peers, and (*ii*) minimizing the number of peers that are involved for each search request. To this end, ISM is composed by: (*i*) a *Profile Mechanism*, according to which each peer builds a "profile" for each of its neighboring peer; (*ii*) a *Query Similarity* function, which calculates the similarity queries to a new query; (*iii*) a *Relevance Rank*, which is a ranking technique for peers that takes as input the (neighboring) peer profiles, and produces as output a ranked list of (neighboring) peers used to bias the search towards the most relevant peers; and (*iv*) a *Search Mechanism*, which implements the ISM search policy. In [52], authors show how ISM works well (*i*) when peers hold some specialized knowledge about the P2P environment, and (*ii*) over P2P networks having high degrees of query locality; in these particular conditions, ISM outperforms BFS as well as RBFS techniques. Other techniques that can be classified as belonging to the IST class are: the framework proposed by Gen Yee and Frieder [16], which combines ranking techniques and metadata management for efficiently supporting IR over P2P networks; the *Metasearch Engines* by Meng *et al.* [33], mainly focused on source selection and merging of results from independent sources; the system proposed by Sripanidkulchai *et al.* [44], which discriminates among peers based on their past behavior in order to form communities of peers having similar interests. The main limitation of IST is that, being usually implemented inside the P2P middleware directly, they could become resource-consuming and, in general, do not scale well on large and dynamic P2P networks.

4.4 Statistics-based Search Techniques

Statistics-based Search Techniques (SST) are another important result in the context of querying P2P networks: they use some aggregated statistics to forward queries towards a particular sub-set of peers; usually, statistics is maintained by mining results of past queries. Techniques belonging to such a class are: (*i*) the *Most Results in Past* (>RES) heuristic, proposed by Yang and Garcia-Molina [50], where query messages are routed to those peers that returned the most results for the last *m* queries, being *m* a technique parameter (it should be noted that, in this case, the statistics employed is very simple being based on a "quantitative" approach); (*ii*) Galanis *et al.*'s data summaries and histograms [14], which are built on each peer by means of data replication techniques, in order to exploit such information at query time to bias the search towards the most relevant peers; (*iii*) Gong *et al.*'sbloom filters [20], and Koloniari and Pitoura's multi-level bloom filters [28], which are statistics-inspired compressed data structures able to summarize the data of the neighborhood of a given peer, and to guide query answering over P2P networks over P2P networks by means of probabilistic algorithms. Even if statistics allows the performance of certain kinds of queries (e.g., range queries [22]) to be improved when compared against traditional approaches, maintaining summary data structures over P2P networks can become very resource-intensive thus unfeasible in real-life scenarios.

4.5 Index-based Search Techniques

Index-based Search Techniques (IndST) efficiently exploit the hierarchical nature of structured P2P networks, and extensively use and take advantages from well-known data indexing solutions coming from the RDBMS technology (e.g., B^+-trees and *R*-trees). Among IndST proposals, we recall: (*i*) Clarke *et al.*'s *Freenet* [7], which uses an intelligent indexing scheme based on the *Depth-First-Search* (DFS) mechanism to locate objects (note that Freenet has not been proposed for the context of P2P networks properly, but, indeed, for a general "anonymous" information storage and retrieval system); (*ii*) Aberer's *P-Grid* [1], which provides extensions to traditional DHTs in order to efficiently support scalability, which is recognized as one of the critical factors for P2P systems; (*iii*) Ratnasamy *et al.*'s CAN [38], which employs a *d*-dimensional Cartesian space to index resources on P2P networks; (*iv*) Stoica *et al.*'s *Chord* [45], which uses a linear space of identifiers forming a ring, being such ring exploited to speed-up lookup operations over P2P networks; (*v*) Zhao *et al.*'s Tapestry [54], mainly focused on supporting fault-tolerant functionalities over large P2P networks; (*vi*) the hybrid technique for building and maintaining local indices proposed by Crespo and Garcia-Molina [9], that present three different methods (namely, *Compound Routing Index* (CRI), *Hop-Count Routing Index* (HRI), and *Exponentially Aggregated Routing* Index (ERI)) which generate indexing data structures able to store the "direction" towards relevant documents over P2P networks, by exploiting original ideas about routing protocols developed and tested in *Arpanet*; (*vii*) Gibbons *et al.*'s *IrisNet* [17], which is an architecture for supporting IR over P2P networks storing XML data indexed by XPath expressions; (*viii*) the *Locality Sensing Hashing* (LSH) technique by Gupta *et al.* [22], that propose using horizontal partitions of relational tables and extensions to the original DHT for supporting approximate range query answering over P2P networks; (*ix*) the Bremer and Gertz's distributed indexes (namely, *P*-index, *A*-index, and *T*-index) [4], which are oriented to build virtual XML repositories over P2P networks by encoding global path information, and efficiently support global query answering over them; (*x*) *P-Trees* by Crainiceanu *et al.* [8], that propose augmenting the DHT with B^+-trees in order to support range query answering on relational P2P databases; (*xi*) Loo *et al.*'s *PierSearch* [30], which exports RDBMS features in an Internet-scale P2P environment; (*xii*) Sartiani *et al.*'s *XPeer* [42], which is targeted at XML data, and uses full tree-guides to perform query evaluation; (*xiii*) *XP2P*, proposed by us in [2; 3], where lightweight XPath expressions are encoded in few KB by means of Rabin's fingerprints in order to build a (lightweight) distributed index for efficiently supporting lookup queries over structured P2P networks. Just like SST, even if query capabilities are improved and well-supported, also including new paradigms that were missing in first-generation P2P systems (such as range queries), IndST mainly suffer from scalability limitations, and updating distributed indexes over large P2P networks is still an open problem.

4.6 Distributed Information Retrieval Techniques

The *Distributed Information Retrieval* (DIR) approach [6; 15; 37; 49], first proposed in contexts different from P2P systems, assumes that peers have a *global knowledge* of the system, e.g. statistical knowledge about the content of *each* data repository in the network, or intensional (i.e., schema-based) knowledge as in [5; 22; 23; 36; 46]. By contrary, most of actual P2P query techniques, which are the baseline for even-complex P2P IR methodologies, assume that peers only have a *local knowledge* of the system, e.g. *knowledge about their neighboring peers*. This is mainly due to the fact that *Distributed Information Retrieval Techniques* (DIRT), being based on the assumption of holding (and maintaining) global views of the system through peers, could become very inefficient in large and dynamic P2P networks.

4.7 Semantics-based Search Techniques

Semantics-based Search Techniques (SemST) are the new frontier in the context of querying P2P networks. Such techniques aim at adopting formal semantics to both model and query distributed resources over P2P networks, in order to improve the capabilities of traditional resource-sharing P2P systems. The first advantage of SemST is the amenity of re-using and re-adopting well-founded results coming from semantic models and query languages. Another advantage consists in the possibility of meaningfully integrating query techniques in P2P networks with leading new research trends like *Ontologies* and *Semantic Web*. In literature, there are very few proposals addressing the described research challenges, since most papers are still focused on query performances of unstructured and structured P2P systems. However, it is expected that integrating semantics in P2P networks will be one of the most relevant research topic for next-generation data-intensive applications.

First, Crespo and Garcia-Molina propose the notion of *Semantics Overlay Networks* (SON) [Crespo and Garcia-Molina, 2003], which are an efficient way of grouping together peers that share the same schema information. Therefore, peers having one or more topics on the same *thematic hierarchy* belong to the same SON. This approach well-supports query routing as every peer p_i can quickly identify peers containing relevant information, namely the set $N(p_i)$, by avoiding network flooding. Here, "relevant" means that a certain semantic relation exists between information held in p_i and information held in peers belonging to $N(p_i)$. Such semantic hierarchies are naturally represented (and processed) via the *Resource Description Framework* (RDF), by also taking advantages from several declarative languages for querying and defining views over RDF bases, such as *RDF Query Language* (RQL) [25] and *RDF View Language* (RVL) [32]. In [9], authors demonstrate that SON can significantly improve query performances while at the same time allowing users to decide what content to publish in their (peer) hosts, i.e. how to form a SON. The SON initiative heavily influenced many P2P-focused research projects; among all, some of them are centered on query routing issues in SON, by meaningfully using the potentialities of RDF constructs: *RDFPeers* by Cai and Frank [5], *SemDIS* by Halaschek *et al.* [22], *Piazza* by Halevy *et al.* [23], *Edutella* by Nejdl *et al.* [36], and *Self-Organizing SON* by Tang *et al.* [46]. All these initiatives have in common the idea of propagating queries across a (semantic) P2P system by means of semantics-based techniques such as correlation discovery, containment etc, mainly working on RDF-modeled networks of concepts.Another significant proposal can be found in the *ICS-FORTH SQPeer Middleware* [27], proposed by Kokkinidis and Christophides, that, starting from the same motivations of the SON initiative, propose more sophisticated information representation and extraction mechanisms by introducing (*i*) the concept of *active RDF schemas* for declaring the parts of a SON RDF schema that are *currently* of interest for a peer, via an advertisement mechanism, and (*ii*) the concept of *semantic query pattern* relying on query/view subsumption techniques that are able to guide query routing on the basis of semantics.

A possible limitation of SON is represented by the overwhelming volume of messages that can be generated for supporting data object replications on peers, as required by SON design guidelines [10]. Thus, P2P applications running on top of the SON-based model for query routing incur excessive overheads on network traffic. An interesting solution to this problem has been proposed by Li *et al.* [29]: they suggest using *signatures on neighboring peers* for directing searches along selected *network paths*, and introduce some schemes to facilitate efficient search of data objects. Signatures are a way of adding semantics to data, by building a bit vector *V*; *V* is generated according to the following two steps: (*i*) hash the content of a data object into bit strings, said *BS*, and (*ii*) apply a bitwise OR operator on *BS*. The so-built bit strings are used at query time by performing a bitwise AND operation on the *search signature* (i.e., the signature of the term used as search key) and the *data signature* (i.e., the signature stored on the *current* peer). In [29], authors show how some proposed flooding-based search algorithms allow the signatures of the neighboring peers to be efficiently exploited to enhance search results, and, in addition to this, an extensive experimental part clearly confirms the effectiveness of the neighborhood signature technique in comparison with existing P2P content search methods, including Gnutella [18], RWA [31], and the IndST of the systems *Morpheus* [34] and *OceanStore* [39].

Latent Semantic Indexing (LSI), first proposed by Deerwester *et al.* [13], is a state-of-the-art technique for supporting IR from collections of documents. It is based on the *Vector Space Model* (VSM), proposed by Salton *et al.* [Salton et al., 1975], which represents a document collection as a term-by-document matrix $A_{t´d}$ such that each element of $A_{t´d}$ is a weight w_{ij} of the corresponding term t_i in the specific document d_j; w_{ij} is computed by taking into account (*i*) the term frequency tf_{ij}, which captures the local relevance of t_i in d_j, and (*ii*) the inverted document frequency df_j, which models a sort of global statistic knowledge about d_j in the whole document collection [41]. As opposite to the original VSM approach, the main idea behind LSI is comparing user's queries and documents at the concept level (thus using semantics) rather than on the basis of simple keyword matching (as in VSM). In LSI, the VSM matrix $A_{t´d}$ is factorized by the *Single Value Decomposition* (SVD) technique, first proposed by Golub and Loan [19], which, unfortunately, introduces an excessive computational cost when applied on huge document collections, as studied by Zhang *et al.* [53]. However, similarly to DIRT, LSI requires that each peer/site holds global knowledge about the system, thus it becomes ineffective for large P2P networks. A possible solution to this problem can be found in the work of Zhu *et al.* [55], that propose a novel query optimization scheme, called *Semantic Dual Query Expansion* (SDQE), where the lack of global knowledge is compensated by engaging ad-hoc query optimization plans implemented on peers locally. SDQE is in turn based on the *Query Expansion* (QE) technique, initially proposed by Salton and Buckley [40],which consists in refining user's queries by adding to them other terms related to those expressed by the original queries. Some works in the context of DIRT (e.g., [15; 37; 49]) show that DIRT performance can be improved thanks to QE. In other words, SDQE can be just considered as a sort of advanced query engine for DIRT, which allows us to mitigate the requirement of global knowledge posed by DIRT.

Finally, in [11;12], we propose an innovative *semantics-based framework* for supporting KD- and IR-style resource querying on large scale P2P XML repositories (e.g., those that one can find in corporate B2B and B2C *e*-commerce systems). In more detail, in [12], we propose (*i*) modeling both XML repositories/documents and queries in terms of concepts they express by means of formal reasoning flat models, like lists, and hierarchical models, like graphs, and (*ii*) applying ad-hoc knowledge extraction algorithms that efficiently exploit such models. Basically, these intuitions lead to the definition of the so-called *Semantic Communities of Peers* (SCoP), which connect peers storing semantically-related information, and are used at query time to efficiently extract knowledge from peers. SCoP are built and maintained on the basis of the analysis of results of past user's queries through the neighboring peers (i.e., the local knowledge), without considering neither (*i*) any pre-fixed scheme, which is usually set by the system author, and, thus, could not fit the "real" evolution of the target, nor (*ii*) intensional models (i.e., schema-based reasoning), like in SON and SQPeer, which do not take into account query result analysis. On the contrary, in some sense, SCoP evolve naturally over time, according to the analysis of query results, i.e. user activities against the system, thus meaningfully taking advantage from the local knowledge about neighboring peers. Specifically, algorithms presented in [12] enhance the semantic expressiveness of the reasoning task via exploiting the local knowledge given by mining, according to some meaningful two-dimensional abstractions, past (successful) query results flooded through neighboring peers.

5. A Comparative Analysis of P2P Query Strategies

As demonstrated in Section 4, there is a wide and heterogeneous literature on the issue of efficiently querying P2P networks. In this Section, we provide a comparative analysis of state-of-the-art query strategies for such networks.

Basically, BST and RST are the first experiences in this research field. Search activities developed on top of these techniques propagate the query message from a peer to another one by flooding the network according to a spanning-oriented approach. RST is different from BST in the fact that it selects next peers to be accessed during query evaluation at random. The main characteristic of these strategies consists in assuming the lack of a global mediate (data)schema among peers, so that it is not possible to develop specialized search tasks taking advantages from the a-priori knowledge about schemas (i.e., the knowledge that can be retrieved at query time before to access the peer database). On the other hand, although query performance of these techniques is low, they do not require particular data structures to be implemented on peers, so that the spatial complexity is low accordingly.

IST can be regarded as the "natural" evolution of BST and RST. In IST, search activities successfully exploit well-known IR-inspired methodologies in order to extract knowledge from peers, and bias the search towards relevant peers/segment-of-the-network. Although query performance are decisively improved with respect to the capabilities of BST and RST, the main limitation of IST is represented by scalability issues, as IR methodologies require complex data structures to beimplemented on peers, and multi-step algorithms used to retrieve knowledge.

SBT come from the wide literature on computing data summaries from massive amounts of data sets, such as histogram- and sampling-based synopsis data structures, with the novelty of using these summarizing data structures as a solution for biasing the search towards relevant peers. In some sense, they overcome the original meaning of data summaries, and propose using them as service data structures exploited during query evaluation. However, due to important limitations arising from incompleteness of peer information, it still has not been proved the effective feasibility of these query strategies on real-life P2P networks.

IndST are promising as founding on well-understood indexing methodologies coming from RDBMS technologies, which have been successfully applied since many years and still represent effective and efficient solutions to the annoying problem of accessing and querying massive data sets. Performance is good on structured and schema-based P2P networks. However, maintaining distributed indexes across peers is a very problematic issue, which still demands for efficient solutions. In addition to this, the problem of defining specialized indexing data structures in high-rate P2P networks is very attracting and will conquest the research scene during next years.

DIRT are an attempt to re-use DIR methodologies for DBMS in the context of querying P2P networks. Despite the main idea is reasonable, the implementation of DIR techniques on peers in their original versions poses important limitations concerning both spatial and temporal complexities. In fact, latest approaches aim at devising "optimized" versions of these techniques, capable of dealing with innovative requirements introduced by P2P networks.

Contrarily to the above-presented query strategies, in order to discuss on SemST we must focus on specialized techniques, since each technique introduces particular features that demand for an ad-hoc analysis. Following this assumption, here we consider four relevant techniques belonging to the SemST class: SON [10], SQPeer [27], the neighborhood signature technique by Li *et al.* [29], and our technique [12]. With respect to the specific case of devising a comparative analysis of SemST, the goal is to put in evidence the semantic expressiveness of target techniques by also looking at scalability and maintenance issues, which play a critical role in large and dynamic P2P networks.

When compared against constructs and data structures of SON and SQPeer, SCoP retain the important advantage of being a "low cost" solution. In fact, while RDF schemas allow the semantic expressiveness of the modeling and query phases to be significantly improved (specifically, this amenity is feasible thanks to the solution of declaring RDF views on peer communities), system scalability results to be sensitively reduced because of schema and view replication across peers. Besides this, maintaining RDF schemas and views can become very resource-intensive due to specific properties of P2P systems, such as (quickly) changes in the topology, and strong delocalization of peers. Contrarily to the above case, SCoP are represented and managed in a very efficient way, thus preserving network scalability, while at the same time ensuring a higher semantic expressiveness, as our framework [12] also supports the mechanism of defining views over a single SCoP or a SCoP domain, in a similar way to the SQPeer proposal.

When compared with the Li et al.'s neighborhood signature technique, SCoP semantic expressiveness result to be higher, as search mechanisms supported by neighborhood signatures are efficiently but poor on the semantic level (in fact, there is not any semantic-based computational model besides simple bitwise operators), whereas update/maintenance overheads are low in both techniques.

6. Conclusions and Future Research Directions

Querying P2P systems, which in this paper has been investigated as a fundamental task for supporting advanced functionalities such as KD- and IR-style data and resource extraction, is an important research challenge for the Database research community, still asking for solutions capable of capturing all the (complex) requirements posed by such novel class of systems. As highlighted throughout the paper, this challenge involves various aspects ranging from representation and reasoning issues to maintenance and query issues. At the same time, research topics discussed in this paper play a leading role with respect to the efficiency and the effectiveness of a wide range of modern applications, which also have a relevant impact on day-to-day real-life (e.g., *e*-government systems, *e*-procurement systems etc). Given these considerations, we presented a survey of models, issues and algorithms related to the problem of efficiently querying P2P networks. Also, we provided a meaningful taxonomy of state-of-the-art query strategies for P2P networks that puts in evidence current research trends in supporting information and knowledge extraction in large, dynamic, data-intensive P2P systems.

Adding *preferences* into query languages seems to be the next challenge for P2P research. By means of preferences, users/applications can specify *constraints* over the final result, thus designing new paradigms for information and knowledge delivery over P2P networks. As an example, a user could request the list reporting administrative data on Central Europe suppliers that never sold parts in Italy, while, at the same time, never imported parts from East Europe. In order to efficiently support *preference-aware query functionalities*, next-generation P2P systems must include inside their query layers novel models and algorithms capable of be aware about the concept “preference” on the semantic plan as well as the query execution plan, thus overcoming limitations of actual P2P query engines. A possible solution could be obtained by integrating *logical models and languages* with consolidated P2P query routing algorithms, thus taking advantages in terms of flexibility and expressiveness of both the query definition and evaluation tasks.

Another interesting line of research for next-generation P2P query techniques seems to be the *probabilistic dataprocessing* initiative, which aims at embedding *probability models and schemes* inside current data representation and query evaluation solutions for P2P networks. This would allow us to meaningfully extend the capabilities of state-of-the-art P2P systems by (*i*) including novel query paradigms, and (*ii*) providing support for the querying of incomplete/inconsistent data and information sources, which are very popular in very-large and highly-dynamic P2P networks.

References

[1] Aberer, K (2001). P-Grid: A Self-Organizing Access Structure for P2P Information Systems. *In: Proceedings of the 6th International Conference on Cooperative Information Systems*, 179-194.

[2] Bonifati, A., Cuzzocrea, A (2006). Storing and Retrieving XPath Fragments in Structured P2P Networks. *Data & Knowledge Engineering*, 59 (2) 247-269.

[3] Bonifati, A., Cuzzocrea, A., Matrangolo, U., Jain, M (2004). XPath Lookup Queries in P2P Networks. *In: Proceedings of the 6th ACM International Workshop on Web Information and Data Management*, in conjunction with the *13th ACM International Conference on Information and Knowledge Management*, 48-55.

[4] Bremer, J.-M., Gertz, M (2003). On Distributing XML Repositories. In *Proceedings of the 2003 ACM International Workshop on Web and Databases*, in conjunction with the *2003 ACM International Conference on Management of Data*, 73-78.

[5] Cai, M., Frank, M (2004). RDFPeers: A Scalable Distributed RDF Repository based on a Structured Peer-to-Peer Network. *In: Proceedings of the 13th International World Wide Web Conference*, 650-657.

[6] Callan, J (2000). Distributed Information Retrieval. *Advances in Information Retrieval*, Kluwer Academic Publishers, 127-150.

[7] Clarke, I., Sandberg, O., Wiley, B., Hong, T.W (2000). Freenet: A Distributed Anonymous Information Storage and Retrieval System. *In: Proceedings of the ICSI Workshop on Design Issues in Anonymity and Unobservability*, 311-320.

[8] Crainiceanu, A., Linga, P., Gehrke, J., Shanmugasundaram, J (2004). Querying Peer-to-Peer Networks Using P-Trees. *In: Proceedings of the 2004 ACM International Workshop on Web and Databases*, in conjunction with the *2004 ACM International Conference on Management of Data*, 25-30.

[9] Crespo, A., Garcia-Molina, H (2002). Routing Indices For Peer-to-Peer Systems. In *Proceedings of the 22nd IEEE International Conference on Distributed Computing Systems*, 23-34.

[10] Crespo, A., Garcia-Molina, H. (2003). Semantic Overlay Networks for P2P Systems. *Stanford Technical Report, Computer Science Department*, Stanford University.

[11] Cuzzocrea, A (2005). Towards a Semantics-based Framework for KD- and IR-style Resource Querying on XML-based P2P Information Systems. *In: Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, 58-61.

[12} Cuzzocrea, A (2006). On Semantically-Augmented XML-based P2P Information Systems. *In: Proceedings of the 7th International Conference on Flexible Query Answering Systems*, LNAI Vol. 4027, 441-457.

[13] Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R. (1999). Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, Vol. 41, No. 6, 391-407.

[14] Galanis, L., Wang, Y., Jeffery, S.R., DeWitt, D.J. (2003). Locating Data Sources in Large Distributed Systems. *In: Proceedings of the 29th International Conference on Very Large Data Bases*, 874-885.

[15] Gauch, S., Wang, J., Rachakonda, S.M (1999). A Corpus Analysis Approach for Automatic Query Expansion and its Extension to Multiple Databases. *ACM Transactions on Information Systems*, 17 (3) 250-269.

[16] Gen Yee, W., Frieder, O (2005). On Search in Peer-to-Peer File Sharing Systems. In *Proceedings of the 20th ACM Symposium on Applied Computing*, 1023-1030.

[17] Gibbons, P.B., Karp, B., Ke, Y., Nath, S., Seshan, S. (2003). IrisNet: An Architecture for a World-Wide Sensor Web. *IEEE Pervasive Computing*, 2 (4) 22-33.

[18] The Gnutella File Sharing System (2006). http://gnutella.wego.com

[19] Golub, G.H., Loan, C.F.V. (1996). *Matrix Computation*, The John Hopkins University Press.

[20] Gong, X., Yan, Y., Qian, W., Zhou, A. (2005). Bloom Filter-based XML Packets Filtering for Millions of Path Queries. *In: Proceedings of the 21st IEEE International Conference on Data Engineering*, 890-901.

[21] Gummadi, P.K., Dunn, R.J., Saroiu, S., Gribble, S.D., Levy, H.M., Zahorjan, J. (2003). Measurement, Modeling, and Analysis of a Peer-to-Peer File-Sharing Workload. *In: Proceedings of the 19th ACM Symposium on Operating Systems Principles*, 314-329.

[22] Gupta, A., Agrawal, D., El Abbadi, A. (2003). Approximate Range Selection Queries in Peer-to-Peer Systems. *In: Proceedings of the 1st Biennial Conference on Innovative Data Systems Research*, http://www-db.cs.wisc.edu/cidr/cidr2003/program/p13.pdf

[22] Halaschek, C., Aleman-Meza, B., Arpinar, I.B., Sheth, A.P. (2004). Discovering and Ranking Semantic Associations over a Large RDF Metabase. *In: Proceedings of the 30th International Conference on Very Large Data Bases*, 1317-1320.

[23] Halevy, A.Y., Ives, Z.G., Mork, P., Tatarinov, I. (2003). Piazza: Data Management Infrastructure for Semantic Web Applications. *In: Proceedings of the 12th International World Wide Web Conference*, 556-567.

[24] Kalogeraki, V., Gunopulos, D., Zeinalipour-Yazti, D. (2002). A Local Search Mechanism for Peer-to-Peer Networks. *In: Proceedings of the 11th ACM International Conference on Information and Knowledge Management*, 300-307.

[25] Karvounarakis, G., Alexaki, S., Christophides, V., Plexousakis, D., Scholl, M. (2002). RQL: A Declarative Query Language for RDF. *In: Proceedings of the 11th International World Wide Web Conference*, 592-603.

[26] The KaZaA File Sharing System (2006). http://www.kazaa.com.

[27] Kokkinidis, G., Christophides, V. (2004). Semantic Query Routing and Processing in P2P Database Systems: The ICS-FORTH SQPeer Middleware. *In: Proceedings of the 2004 International Workshop on Peer-to-Peer Computing and Databases*, in conjunction with the *9th International Conference on Extending Database Technology*, 486-495.

[28] Koloniari, G., Pitoura, E (2004). Content-Based Routing of Path Queries in Peer-to-Peer Systems. *In: Proceedings of the 9th International Conference on Extending Database Technology*, 29-47.

[29] Li, M., Lee, W.-C., Sivasubramaniam, A. (2003). Neighborhood Signatures for Searching P2P Networks. *In: Proceedings of the 7th IEEE International Database Engineering and Applications Symposium*, 149-159.

[30] Loo, B.T., Huebsch, R., Hellerstein, J.M., Stoica, I., Shenker, S (2004). Enhancing P2P File-Sharing with an Internet-Scale Query Processor. *In: Proceedings of the 30th International Conference on Very Large Data Bases*, 432-443.

[31] Lv, Q., Cao, P., Cohen, E., Li, K., Shenker, S. (2002). Search and Replication in Unstructured Peer-to-Peer Networks. *In: Proceedings of the 16th ACM International Conference on Supercomputing*, 84-95.

[32] Magkanaraki, A., Tannen, V., Christophides, V., Plexousakis, D. (2003). Viewing the Semantic Web Through RVL Lenses. *In: Proceedings of the 2nd International Semantic Web Conference*, 96-112.

[33] Meng, W., Yu, C., Liu, K.-L. (2002). Building Efficient and Effective Metasearch Engines. *ACM Computing Surveys*, 34 (1) 48-84.

[34] Morpheus File Sharing System. http://www.musiccity.com.

[35] The Napster File Sharing System (2006). http://www.napster.com

[36] Nejdl, W., Wolpers, M., Siberski, W., Schmitz, C., Schlosser, M., Brunkhorst, I., and Loser, A. (2003). Super-Peer-based Routing and Clustering Strategies for RDF-based P2P Networks. *In: Proceedings of the 12th International World Wide Web Conference*, 536-543.

[37] Ogilvie, P., Callan, J. (2001). The Effectiveness of Query Expansion for Distributed Information Retrieval. *In: Proceedings of the 10th ACM International Conference on Information and Knowledge Management*, 183-190.

[38] Ratnasamy, P., Francis, P., Handley, M., Karp, R., Shenker, S. (2001). A Scalable Content-Addressable Network. *In: Proceedings of the 2001 ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, 161-172.

[39] Rhea, S., Wells, C., Eaton, P., Geels, D., Zhao, B., Weatherspoon, H., Kubiatowicz, J. (2001). Maintenance-free Global Data Storage. *IEEE Internet Computing*, 5 (5) 40-49.

[40] Salton, G., Buckley, C. (1990). Improving Retrieval Performance by Relevance Feedback. *Journal of the American Society for Information Science*, 41 (4) 288-297.

[41] Salton, G., Wang, A., Yang, C.S. (1975). A Vector Space Model for Information Retrieval. *Journal of American Society for Information Science*, 18 (11) 613-620.

[42] Sartiani, C., Manghi, P., Ghelli, G., Conforti, G. (2004). XPeer: A Self-Organizing XML P2P Database System. *In: Proceedings of the 2004 International Workshop on Peer-to-Peer Computing and Databases*, in conjunction with the *9th International Conference on Extending Database Technology*, 456-465.

[43] Schmidt, A., Waas, F., Kersten, M., Carey, M., Manolescu, I., Busse, R. (2002). XMark: A Benchmark for XML Data Management. *In: Proceedings of the 28th International Conference on Very Large Data Bases*, 974-985.

[44] Sripanidkulchai, K., Maggs, B., and Zhang, H. (2003). Efficient Content Location using Interest-based Locality in Peer-to-Peer Systems. *In: Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies*, 81-87.

[45] Stoica, I., Morris, R., Karger, D., Frans Kaashoek, M, Balakrishnan, H. (2001). Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications. *In: Proceedings of the 2001 ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, 149-160.

[46] Tang, C., Xu, Z., Dwarkadas, S (2003). Peer-to-Peer Information Retrieval using Self-Organizing Semantic Overlay Networks. *In: Proceedings of the 2003 ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, 175-186.

[47] Tsoumakos, D., Roussopoulos, N (2003a). A Comparison of Peer-to-Peer Search Methods. *In: Proceedings of the 2003 ACM International Workshop on Web and Databases*, in conjunction with the *2003 ACM International Conference on Management of Data*, 61-66.

[48] Tsoumakos, D., Roussopoulos, N (2003b). Adaptive Probabilistic Search for Peer-to-Peer Networks. *In: Proceedings of the 3rd IEEE International Conference on Peer-to-Peer Computing*, 102-109.

[49] Xu, J., Callan, J (1998). Effective Retrieval with Distributed Collections. *In: Proceedings of the 21st ACM International*

Conference on Research and Development in Information Retrieval, 112-120.

[50] Yang, B., Garcia-Molina, H (2002). Efficient Search in Peer-to-Peer Networks. *In: Proceedings of the 22nd IEEE International Conference on Distributed Computing Systems*, 5-14.

[51] Zeinalipour-Yazti, D., Kalogeraki, V., Gunopulos, D. (2004). Information Retrieval Techniques for Peer-to-Peer Networks. *IEEE CiSE Magazine, Special Issue on Web Engineering*, 12-20.

[52] Zeinalipour-Yazti, D., Kalogeraki, V., Gunopulos, D. (2005). Exploiting Locality for Scalable Information Retrieval in Peer-to-Peer Systems. *Information Systems*, 30 (4) 277-298.

[53] Zhang, X., Berry, M.W., Raghavan, P (2001). Level Search Schemes for Information Filtering and Retrieval. *Information Processing and Management*, 37 (2) 313-334.

[54] Zhao, Y.B., Kubiatowicz, J., Joseph, A (2001). Tapestry: An Infrastructure for Fault-Tolerant Wide-Area Location and Routing. *Technical Report UCB/CSD-01-1141, Computer Science Division*, U. C. Berkeley.

[55] Zhu, X., Cao, H., Yu, Y (2006). SDQE: Towards Automatic Semantic Query Optimization in P2P Systems. *Information Processing and Management*, 42 (1) 222-236.

Alfredo Cuzzocrea received the Laurea Degree in Computer Science Engineering on April 2001 and the PhD Degree in Computer Science and Systems Engineering on February 2005, both from the University of Calabria. Presently, he is a Researcher at the Institute of High Performance Computing and Networking of the Italian National Research Council, and Contract Professor at the Department of Electronics, Computer Science and Systems of the University of Calabria, where he is a member of the Database Research Group. His research interests include multidimensional data modeling and querying, data stream modeling and querying, data warehousing and OLAP, XML data management, Web information systems modeling and engineering, knowledge representation and management models and techniques, Grid and P2P computing. He is author or co-author of more than 60 papers in referred international conferences (including SSDBM, DEXA, DaWaK, DOLAP, IDEAS, SEKE, WISE, FQAS, SAC) and journals (including JIIS, DKE, WIAS). He serves as program committee member of referred international conferences (including ICDM, CIKM, PAKDD, DaWaK, DOLAP, SAC) and as review board member of referred international journals (including TODS, TKDE, INS, IJSEKE, FGCS, JDIM). Up-to-date information is available at http://si.deis.unical.it/~cuzzocrea