Design Issues and Challenges for RDF- and Schema-Based Peer-to-Peer Systems

Wolfgang Nejdl* and Wolf Siberski* and Michael Sintek†

Abstract

Databases have employed a schema-based approach to store and retrieve structured data for decades. For peer-to-peer (P2P) networks, similar approaches are just beginning to emerge. While quite a few database techniques can be re-used in this new context, a P2P data management infrastructure poses additional challenges which have to be solved before schema-based P2P networks become as common as schema-based databases. We will describe some of these challenges and discuss approaches to solve them. Our discussion will be based on the design decisions we have employed in our Edutella infrastructure, a schema-based P2P network based on RDF and RDF schemas, and will also point out additional work addressing the issues discussed.

1 Introduction

P2P applications have been quite successful for exchanging music files, where networks use simple attributes to describe these resources. A lot of effort has been put into refining topologies and query routing functionalities of these networks, and simple systems like Napster and Gnutella have inspired more efficient infrastructures e.g. based on distributed hash tables (e.g. CAN and CHORD [27, 32]). Less effort has been put into extending the representation and query functionalities offered by such networks, projects exploring more expressive P2P infrastructures [23, 3, 2, 17] have only slowly started the move toward schema-based P2P networks.

At the same time, database systems have evolved toward a higher degree of distribution. While it has been a long way from central databases to truly distributed databases, we currently see first explorations toward true peer-to-peer data management infrastructures which will have all characteristics of P2P systems, i.e. *local control of data*, *dynamic addition and removal of peers*, *only local knowledge of available data and schemas* and *self-organization and -optimization*.

In this view, schema-based P2P systems are the point where these two directions of research meet (see also [16]), as shown in figure 1.[1]

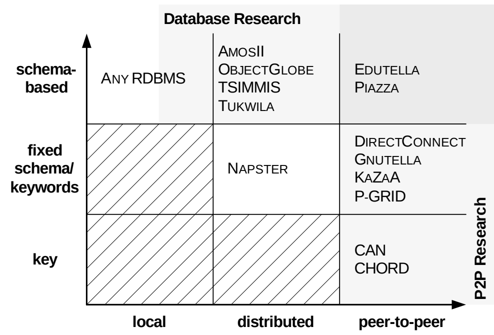

Figure 1: Schema Capabilities and Distribution

A third interesting dimension relevant for schema-based P2P networks is *homogeneity*, addressing the need for heterogeneous schemas and for providing mappings between them.

In the Edutella project [12, 23, 24] we have been exploring some of these issues, with the goal of designing and implementing a schema-based P2P infrastructure for the Semantic Web. Edutella relies on the W3C metadata standards RDF and RDF Schema [20, 6] to describe distributed resources, and uses basic P2P primitives provided as part of the JXTA framework [15]. We will describe some important building blocks for the Edutella system and use this discussion as well as references to other related work to illustrate some important design decisions relevant for moving toward schema-based P2P networks.

2 Building Blocks and Challenges

To provide a schema-based peer-to-peer infrastructure, we need at least the following building blocks:

- a schema language to define and use the various schemas which specify the kind of data available in the P2P network
- a query language to retrieve the data stored in the P2P network
- a network topology combined with an appropriate query routing algorithm to allow efficient queries

* {nejdl,siberski}@learninglab.de, Learning Lab Lower Saxony, University of Hannover, Germany

† sintek@dfki.de, DFKI GmbH, Kaiserslautern, Germany

[1] This figure is not meant to be complete, we have just depicted a few systems for illustration purposes.

- facilities to integrate heterogeneous information stored in the P2P network

In the following sections we describe these building blocks, the design decisions we have explored in the context of the Edutella system as well as other related work, and some future research challenges for Edutella in particular and schema-based P2P networks in general.

2.1 Schema Languages

In the Semantic Web, an important aspect for its overall design is the exchange of data among computer systems without the need of explicit consumer-producer relationships. RDF and RDFS [20, 6] are used to annotate resources on the Web and provide the means by which computer systems can exchange and comprehend data. All resources are identifiable by unique resource identifiers (URIs plus anchor ids). Annotations about resources are based on various schemas that are defined in RDFS (and extensions thereof) and are stored in RDF repositories, possibly using more than one schema in a repository.

Another important characteristic of RDF metadata is the ability to use distributed annotations for one and the same resource. In contrast to traditional (non-distributed) database systems, it is not necessary to store all annotations of a resource on one server. One server might store metadata which include properties such as *name* for specific resources using a schema based on the Dublin Core metadata standard. Other servers could hold metadata that provide properties for the same resources, possibly using other metadata standards / schemas. This ability for distributed allocation of metadata makes RDF very suitable for the construction of distributed repositories.

Using RDFS, we can represent schemas based on classes, properties and property constraints, to define the vocabulary used for describing our resources. RDF triples $<subject, property, value>$ represent specific annotations, where $subject$ identifies the resource we want to describe (using a URI), $property$ specifies what property we use, and $value$ the specific value, expressed as a primitive datatype or another URI.

RDF schemas are flexible and extensible and can be evolved over time, e.g. by extending them with additional properties. We can use any properties defined in the schemas we use, possibly mix different schemas, and relate different resources to each other, when we want to express interdependencies between these resources, hierarchical relationships, or others.

2.2 Query Languages

Obviously, RDF- and other schema-based P2P networks need query capabilities going beyond the simple key and keyword-based queries. In the context of the Semantic Web more expressive query formalisms have been investigated, which usually build on rule-like languages [4, 23, 31]. This is necessary because, although RDF data are basically graphs, query languages based on simple graph matching and subgraph extraction are not sufficient: they cannot reason about the semantics underlying such data, given in the form of schema languages like RDFS or OWL [34]. Even if we have a query language that takes RDFS into account, this built-in support for exactly one fixed schema language is not sufficient, as it does not allow us to query and combine RDF data expressed in multiple schema languages which is necessary in the case of distributed scenarios where providers can neither be forced to use the same schema nor the same schema language.

It is therefore required that a query language supports the definition of the semantics of several schema languages. This can appropriately be done with rule languages based on Datalog (or Horn logic in general), e.g., with RDF-QEL in Edutella [23]. A syntactic extension of Horn logic, TRIPLE [31], takes the specifics of RDF (like namespaces, resources, and statements) into account and also allows the definition of parameterized views. In Edutella, RDF-QEL provides us with an expressive query exchange language which serves as a common query interchange format, into which local query languages can be translated (quite a common approach in distributed databases). Edutella peers are connected to the network using a wrapper-based architecture, where the wrapper is responsible for translating local query languages into the Edutella common query model. The specific TRIPLE constructs and especially the parameterized views are extremely useful for providing transformation and mediation functionalities, as we will discuss in section 2.4.

2.3 Network Topologies

The second design aspect we have to explore is how to organize peers such that efficient search is possible. We will start from super-peer networks as a particularly appropriate topology for our schema-based P2P topologies, and discuss topology and routing indices in the Edutella network. We also sketch some first explorations into clustering approaches useful in such environments.

Super-Peer Networks. Not all peers are created equal. As discussed in [35], exploiting the different capabilities in a P2P network can lead to an efficient network architecture, where a small subset of peers, called super-peers, takes over specific responsibilities for peer aggregation, query routing and possibly mediation. The KaZaA network [18] uses a simple version of a super-peer based architecture, more elaborate versions are described in [10] and [25]. Super-peer-based P2P infrastructures usually exploit a two-phase routing architecture, which routes queries first in the super-peer backbone, and then distributes them to the peers connected to the super-peers. The last step can sometimes be avoided if the super-peers cache data from their connected peers. Super-peer routing is usually based on different kinds of indexing and routing tables, as discussed in [10] and [25].

The Edutella Super-Peer Topology. The Edutella super-peers [25] employ routing indices, which explicitly acknowledge the semantic heterogeneity of schema-based P2P networks, and therefore include schema information as well as other possible index information. This super-peer backbone is responsible for message routing and integration / mediation of metadata. Super-peers in the Edutella network are arranged in the HyperCuP topology [30]. The HyperCuP algorithm is capable of organizing peers in a P2P network into a recursive graph structure from the family of Cayley graphs, out of which the hypercube is the most well-known topology. The hypercube topology allows for $\log_2 N$ path length and $\log_2 N$ number of neighbors, where N is the total number of nodes in the network (i.e. the number of super-peers in our case).

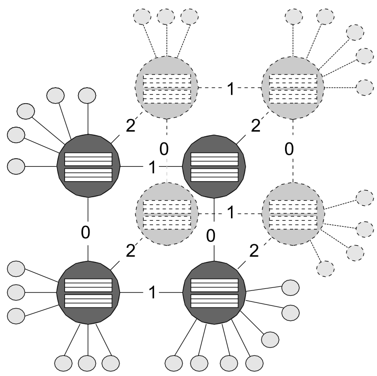

Figure 2: HyperCuP Super-peer Topology

HyperCuP enables efficient query broadcasts and guarantees non-redundant broadcast. The broadcast algorithm works as follows: Each connection is labeled with its dimension in the hypercube. A node invoking a broadcast sends the broadcast message to all its neighbors, tagging it with the edge label on which the message was sent. Nodes receiving the message restrict the forwarding of the message to those links tagged with higher edge labels. This algorithm guarantees that exactly N-1 messages are required to reach all nodes in a topology. Furthermore, the last nodes are reached after $\log_2 N$ forwarding steps.

A new super-peer is able to join the network by asking any other already integrated super-peer which then carries out the peer integration protocol. $O(\log(N))$ messages are sent in order to integrate the new super-peer and maintain a hypercube-like topology (see [30] for a detailed description), irrespective of the number of super-peers currently active in the network.

Peers connect to the super-peers in a star-like fashion, providing content and content metadata, as shown in Figure 2.[2] Alternatives to this topology are possible, as long as they guarantee the spanning tree property for the super-peer backbone, which we exploit for maintaining our routing indices and distributed query plans [7]. Other topologies are possible for other kinds of indices [10]. In contrast to other super-peer networks, Edutella uses designated super-peers, normal peers do not usually include the routing and indexing code used in super-peers (though in principle they could).

Super-Peer/Peer Routing Indices. Edutella super-peers characterize their associated peers using super-peer/peer routing indices. These SP/P indices are updated when a peer connects to a super-peer, and contain all necessary information about connected peers. Entries are valid only for a certain time, and are deleted when the peer does not renew / update it regularly (e.g. because it leaves the network). Peers notify the super-peer when their content changes in ways that trigger an update of the index. In contrast to other approaches (Gnutella [13], CAN [27]), our indices do not refer to individual content elements but to peers (as in CHORD [32]). Indices contain information about peers at different granularities: schema identifiers, schema properties, property value ranges, and individual property values. Future work will focus on identifying optimal granularity levels for different application scenarios.

Super-Peer/Super-Peer Routing Indices. As with peers, we want to avoid broadcasting queries to all super-peers. To achieve this goal we introduce super-peer/super-peer routing indices to route among the super-peers. These SP/SP indices are essentially extracts and summaries from the local SP/P indices. They contain the same kind of information as SP/P indices, but refer to the (direct) neighbors of a super-peer. Queries are forwarded to super-peer neighbors based on the SP/SP indices (restricting the basic HyperCuP broadcast), and sent to connected peers based on the SP/P indices.

Update of Edutella SP/SP indices is based on the registration (or update) messages from connected peers. Whenever an SP/P index changes, this change is propagated to (potentially) all super-peers using a (reversed) HyperCuP broadcast. Whenever an SP/SP index stays the same after the update, propagation stops.

Because one important aspect of P2P networks is their dynamicity, the SP/SP indices are not, in contrast to distributed architectures in the database area (e.g. [5]), replicated versions of a central index, but rather parts of a distributed index similar to routing indices in TCP/IP networks.

Clustering. Query routing in P2P networks benefits from suitable clustering algorithms, where the peers are

[2] This figure shows a very simple example of a hypercube: a cube.

arranged in the P2P network based on their peer characteristics. This is especially true if we employ indices, which are useless if the peers connect randomly to a super-peer. In the context of the Edutella project we are exploring several approaches focusing on different design criteria. We will give a short overview over these approaches, together with pointers to related work. All these approaches still need further exploration and evaluation.

Ontology-based Clustering. In [30], we have explored an approach where peers with identical or similar "interests" with respect to an ontology are grouped in concept clusters which are in turn assigned to a specific logical combination of ontology concepts that describes the peers belonging to the cluster. These concept clusters are organized into a hypercube topology to enable routing to specific concept clusters in the topology. Concept clusters themselves are hypercubes or other Cayley graphs, too. Querying the network works in two routing steps: First, the query is propagated to those concept clusters that contain peers which the query is aiming at. Second, a broadcast is carried out within each of these concept clusters.

Another ontology-based clustering approach has been developed by [33]. In this approach, a Zipf distribution is assumed for the resource popularity. Based on this distribution and a grouping criterion (e.g., belonging to a category in an ontology), clusters of peers are formed.

Rule-based Clustering. In [21], we have explored an approach called *rule-based clustering*. The main idea is to group and register peers in subject specific clusters (SSC) via cluster specific rules. Each super-peer represents such a cluster. A typical cluster may group peers with equal properties, e.g. more static properties like specific query and result schemas, specific domain/IP address ranges, or more dynamic properties, like a minimum number of resources at a peer, average answer time or average number of results.[3] Every cluster provides its own rules, expressing which peers are allowed to join the cluster and which peers are denied to enter the cluster. Typically the super-peer's administrator will define such a policy.[4]

Query-based Clustering. Another, more dynamic way is to take the characteristics of queries into account when deciding on the clustering parameters. As query characteristics in a large and possibly heterogeneous P2P network cannot be defined in advance, we can use frequency counting algorithms on streams to identify the most commonly used items— schemas, properties, or value entries from a taxonomy—in the sent queries. Each peer, super-peer and query is characterized by a set of items, which can be used to cluster similar peers. This means that we have to add a frequency property in our SP/SP routing indices, include the most frequently used items in the indices, and use these statistics to cluster peers to super-peers accordingly. Queries not covered by these (possibly incomplete) SP/SP indices can be broadcast in the super-peer HyperCuP network, and then forwarded to the appropriate peers based on the SP/P indices. The algorithms for estimating the frequencies of specific items discussed in [22], such as sticky sampling and lossy counting, can both be used in this context, as they scan the input stream only once, and do not need much temporary storage to hold the frequency counts.

[3] Further properties for peers and information sources are, for example, discussed in [14]

[4] Some existing single schema P2P networks for file sharing purposes like *Direct Connect* [26] and *E-Donkey* already use simple administrator based rules for clustering peers successfully.

2.4 Information Integration

Finally, we have to integrate data from different peers, and we have to integrate data described by different schemata. While this has been an active research topic for heterogeneous databases, which we can build upon—TSIMMIS [9], for example, makes several assumptions suitable also for the kind of networks we discuss in this paper—highly dynamic P2P networks building only on local knowledge in each peer pose additional challenges worth exploring. We will sketch our ongoing efforts for combining P2P architectures and distributed query processing, as well as for providing mediation and transformation capabilities in the Edutella network, as well as relating them to other work.

Distributed Query Processing. Currently, data processing in P2P networks works as follows: The P2P network is queried for information satisfying some given conditions, and this query is routed to the peers which can answer it. When the possibly large number of results from distinct data sources is returned to the client peer, further query processing takes place centrally at the client. On the other side, data integration systems such as ObjectGlobe [5] distribute query plans to the distributed hosts as much as possible and thus are able to place operators close to the data sources. To generate the query plans, however, these systems need to know where all data are located.

The naive and straightforward way to combine both approaches is to first use P2P capabilities to find out where data is stored, and then use this information to generate a distributed query plan at the client. This query plan becomes instantiated and executed on different hosts. In [7] we explore the interleaving of P2P techniques and query processing instead, by pushing *abstract query plans* through the super-peer-based network, where each super-peer picks and expands those parts of the query plan that can be executed locally. The decision which operations can be executed locally is guided by SP/SP and SP/P indices. This leads to a dynamic *on the fly* distribution and expansion of query plans. This approach enables us to place operators next to data sources and utilize distributed computing resources more effectively.

The expansion of these abstract query plans can be based on different strategies, related to the quality of clustering in the P2P network. If the data are clustered well with respect to the queries, it is most efficient to push joins in the query as near as possible to the data sources, and then take the union of the results for these joins. If the clustering does not reflect the partitions needed by the query, it is more beneficial to gather the data and do the joins on these data on a central super-peer (see also [19]).

Mediation. In contrast to typical file-sharing systems which assume a single global schema for describing their data, Edutella and other schema-based P2P networks [21], [1], [2], [17] have to consider the use of different heterogeneous schemas in the network. As already observed in these papers, instead of using global schemas and correspondences we have to rely on local transformation mechanisms and rules.

In the context of the Edutella project, we have so far explored two lines of research based on two different kinds of mediation peers as part of the Edutella network. Work we have done so far in this context was based on explicit mediation peers as well as rule-based mediation facilities in all super-peers. Extending (restricted) mediation capabilities to all peers such as proposed in [2] and [1] is another promising avenue of research.

Mediation built on AMOS-II. The first line of research builds upon the mediator system AMOS-II [29], which has been used for wrapping many kinds of sources, including XML and web search engines. It recently has been extended to work as a mediator and search engine for Semantic Web metadata, able to query any RDF and RDFS based meta data on the web [28]. This PSELO mediator engine can also act as a query provider to Edutella, general Edutella queries can be sent to the PSELO Edutella provider for evaluation. Currently only RDF and RDFS sources are allowed. The next step is to also allow mediation with relational database sources. This will allow us to process queries that combine web-based educational material with material stored in regular relational databases.

Mediation built on TRIPLE. As has been pointed out in [8], defining views appears to be the right means for mediation, especially in case of schemas or ontologies modeled with the help of description logics. Since on the Semantic Web current approaches for schema/ontology languages build on description logics, e.g., DAML+OIL and its W3C successor OWL [11, 34], a powerful rule language with the capability to define views seems to be a promising candidate for mediation. Especially relevant in our Edutella TRIPLE peer is its capability to define parameterized views which add the flexibility to define multi-step mappings (by nesting/sequencing such views). This peer (currently being developed as part of the ELENA project[5]) allows advanced querying, inferencing and mediation, and also provides reasoning services to be used in ELENA for providing personalization in the context of a smart learning space. It can also be used to express query correspondence assertions and model correspondences as flexible mechanisms to express mappings between heterogeneous schemas as discussed in [21].

3 Conclusion

Schema-based P2P networks and P2P based data management infrastructures build upon traditional P2P networks and distributed / heterogeneous database research, while posing new challenges as well as additional functionalities. This paper has explored some of the design issues relevant for these new schema-based P2P networks, based on the work we have done in the context of the Edutella project as well as on related work.

References

[1] ABERER, K., CUDRÉ-MAUROUX, P., AND HAUSWIRTH, M. The chatty web: Emergent semantics through gossiping. In *Proceedings of the Twelfth International World Wide Web Conference (WWW2003)* (Budapest, Hungary, May 2003).

[2] ABERER, K., CUDRÉ-MAUROUX, P., AND HAUSWIRTH, M. A framework for semantic gossiping. *SIGMOD Record 31*, 4 (December 2002). http://www.p-grid.org/Papers/SIGMOD2002.pdf.

[3] BERNSTEIN, P. A., GIUNCHIGLIA, F., KEMENTSIETSIDIS, A., MYLOPOULOS, J., SERAFINI, L., AND ZAIHRAYEU, I. Data management for peer-to-peer computing: A vision. In *Proceedings of the Fifth International Workshop on the Web and Databases* (Madison, Wisconsin, June 2002).

[4] BOLEY, H., TABET, S., AND WAGNER, G. Design rationale of RuleML: A markup language for semantic web rules. In *Proceedings of the International Semantic Web Working Symposium (SWWS01)* (Stanford, CA, USA, Jul-Aug 2001).

[5] BRAUMANDL, R., KEIDL, M., KEMPER, A., D., K., KREUTZ, A., SELTZSAM, S., AND STOCKER, K. ObjectGlobe: Ubiquitous query processing on the internet. *VLDB Journal: Very Large Data Bases 10*, 1 (2001), 48–71. http://www.db.fmi.uni-passau.de:8000/publications/journals/VLDBJ-ObjectGlobe.pdf.

[6] BRICKLEY, D., AND GUHA, R. V. RDF vocabulary description language 1.0: RDF Schema, Jan. 2003. http://www.w3.org/TR/rdf-schema/.

[7] BRUNKHORST, I., DHRAIEF, H., KEMPER, A., NEJDL, W., AND WIESNER, C. Distributed queries and query optimization in schema-based P2P-systems. Submitted for publication, May 2003.

[5] http://www.learninglab.de/english/projects/elena.html

[8] CALVANESE, D., GIACOMO, G. D., AND LENZERINI, M. Ontology of integration and integration of ontologies. In *Description Logics* (2001).

[9] CHAWATHE, S., GARCIA-MOLINA, H., HAMMER, J., IRELAND, K., PAPAKONSTANTINOU, Y., ULLMAN, J., AND WIDOM, J. The TSIMMIS project: Integration of heterogeneous information sources. In *Proceedings of IPSJ Conference* (Tokyo, Japan, October 1994).

[10] CRESPO, A., AND GARCIA-MOLINA, H. Routing indices for peer-to-peer systems. In *Proceedings International Conference on Distributed Computing Systems* (July 2002).

[11] DAML JOINT COMMITTEE. DAML+OIL, 2001. http://www.daml.org/2001/03/daml+oil-index.html.

[12] The Edutella Project. http://edutella.jxta.org/, 2002.

[13] FRANKEL, J. Gnutella. Web Site - www.gnutella.com, March 1999. Information portal with community, development information and downloads.

[14] GARCIA-MOLINA, H., AND YANG, B. Efficient search in peer-to-peer networks. In *Proceedings of ICDCS* (2002).

[15] GONG, L. Project JXTA: A technology overview. Tech. rep., SUN Microsystems, Apr. 2001. http://www.jxta.org/project/www/docs/TechOverview.pdf.

[16] GRIBBLE, S., HALEVY, A. Y., IVES, Z. G., RODRIG, M., AND SUCIU, D. What can databases do for peer-to-peer. In *Proceedings of the Fourth International Workshop on the Web and Databases (WebDB '2001).* (Santa Barbara, CA, USA, May 2001).

[17] HALEVY, A. Y., IVES, Z. G., MORK, P., AND TATARINOV, I. Piazza: Data management infrastructure for semantic web applications. In *Proceedings of the Twelfth International World Wide Web Conference (WWW2003)* (Budapest, Hungary, May 2003).

[18] HEMMING, N. KaZaA. Web Site - www.kazaa.com.

[19] KEMPER, A., AND WIESNER, C. Hyperqueries: Dynamic distributed query processing on the internet. In *The VLDB Journal* (Rome, Italy, 2001), pp. 551–560.

[20] LASSILA, O., AND SWICK, R. W3C Resource Description Framework model and syntax specification, Feb. 1999. http://www.w3.org/TR/REC-rdf-syntax/.

[21] LÖSER, A., NEJDL, W., WOLPERS, M., AND SIBERSKI, W. Information integration in schema-based peer-to-peer networks. In *Proceedings of the 15th Conference On Advanced Information Systems Engineering (CAISE 03)* (Klagenfurt/Velden, Austria, June 2003).

[22] MANKU, G. S., AND MOTWANI, R. Approximate frequency counts over data streams. In *Proceedings of the 28th International Conference on Very Large Data Bases* (Hong Kong, China, August 2002).

[23] NEJDL, W., WOLF, B., QU, C., DECKER, S., SINTEK, M., NAEVE, A., NILSSON, M., PALMÉR, M., AND RISCH, T. EDUTELLA: a P2P Networking Infrastructure based on RDF. In *Proceedings of the 11th International World Wide Web Conference* (Hawaii, USA, May 2002). http://edutella.jxta.org/reports/edutella-whitepaper.pdf.

[24] NEJDL, W., WOLF, B., STAAB, S., AND TANE, J. Edutella: Searching and annotating resources within an RDF-based P2P network. In *Proceedings of the Semantic Web Workshop, 11th International World Wide Web Conference* (Honolulu, Hawaii, USA, May 2002).

[25] NEJDL, W., WOLPERS, M., SIBERSKI, W., SCHMITZ, C., SCHLOSSER, M., BRUNKHORST, I., AND LOSER, A. Super-peer-based routing and clustering strategies for RDF-based peer-to-peer networks. In *Proceedings of the International World Wide Web Conference* (Budapest, Hungary, May 2003). http://citeseer.nj.nec.com/nejdl02superpeerbased.html.

[26] NEO-MODUS. Direct Connect Homepage. http://www.neo-modus.com/.

[27] RATNASAMY, S., FRANCIS, P., HANDLEY, M., KARP, R., AND SHENKER, S. A scalable content addressable network. In *Proceedings of the 2001 Conference on applications, technologies, architectures, and protocols for computer communications* (2001), ACM Press New York, NY, USA.

[28] RISCH, T. Functional queries to wrapped educational semantic web meta-data. In *Functional Approach to Computing with Data*, P.Gray, L.Kerschberg, P.King, and A.Poulovassilis, Eds. Springer, 2003.

[29] RISCH, T., JOSIFOVSKI, V., AND KATCHAOUNOV, T. Functional data integration in a distributed mediator system. In *Functional Approach to Computing with Data*, P.Gray, L.Kerschberg, P.King, and A.Poulovassilis, Eds. Springer, 2003.

[30] SCHLOSSER, M., SINTEK, M., DECKER, S., AND NEJDL, W. HyperCuP—Hypercubes, Ontologies and Efficient Search on P2P Networks. In *International Workshop on Agents and Peer-to-Peer Computing* (Bologna, Italy, July 2002).

[31] SINTEK, M., AND DECKER, S. TRIPLE—A query, inference, and transformation language for the semantic web. In *Proceedings of the 1st International Semantic Web Conference* (June 2002), Springer LNCS.

[32] STOICA, I., MORRIS, R., KARGER, D., KAASHOEK, M. F., AND BALAKRISHNAN, H. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of the 2001 Conference on applications, technologies, architectures, and protocols for computer communications* (2001), ACM Press New York, NY, USA.

[33] TRIANTAFILLOU, P., XIRUHAKI, C., KOUBARAKIS, M., AND NTARMOS, N. Towards high performance peer-to peer content and resource sharing systems. In *Proceedings of the Conference on Innovative Data Systems Research (CIDR)* (Asilomar, CA, USA, Jan 2003).

[34] W3C WEBONT WORKING GROUP. Web ontology language (OWL), 2003. http://www.w3.org/2001/sw/WebOnt.

[35] YANG, B., AND GARCIA-MOLINA, H. Improving search in peer-to-peer systems. In *Proceedings of the 22nd International Conference on Distributed Computing Systems* (Viena, Austria, July 2002). http://dbpubs.stanford.edu:8090/pub/2001-47.