A CUSTOMIZABLE SEMANTIC-BASED P2P SYSTEM

Marco Mesiti
DICO, Università di Milano, Italy, mesiti@disi.unige.it

Viviana Mascardi
DISI, Università di Genova, mascardi@disi.unige.it

Giovanna Guerrini
DI, Università di Pisa, Italy, guerrini@disi.unige.it

ABSTRACT

Important requirements are nowadays arising in systems for the retrieval of XML documents in P2P networks. Among them we face the problems of service customization and heterogeneity of document structures. Peer willingness to answer queries may be conditioned by a number of factors such as the time the request is received, the characteristics of the peer submitting the query, and the current workload of the peer receiving the query. Thus, appropriate policies should be specified for restricting peer availability to answer queries. Moreover, peers might exploit different structures for representing the same kind of information. Thus, an ontology establishing the mapping among different representations of the same concept is required. In this paper we present a system for the retrieval of XML documents distributed among peers on a hybrid P2P network. Peers are organized in groups and each group contains both a common ontology for representing the documents the group deals with and policies which specify the group and individual peer availability to answer queries.

KEYWORDS

Peer-to-peer, ontology, policy, XML, information retrieval

1. INTRODUCTION

In the last few years a lot of attention has been devoted to peer-to-peer (P2P) systems (D. Milojicic, et al., 2002). The possibility of sharing multimedia information among peers is really attractive as the experience with Napster, Gnutella, and KazaA proved. Current P2P systems focus on handling semantic-free, large-granularity requests for objects by identifier (typically by name), which both limit their usability and restrict the techniques that might be employed to access data (S. Gribble, et al., 2001). Recently, some new approaches have been proposed for content-based retrieval (S. Castano, et al., 2003; I. Klampanos and J. Jose, 2003; G. Koloniari and E. Pitoura, 2004; W. Nejdl, et al., 2003) based on Super-Peer Networks (B. Yang and H. Garcia-Molina, 2003). The adoption of P2P systems to act as huge, flexible, and distributed repositories of XML documents over the Internet is also being investigated (C. Sartiani, et al. 2004).

Current systems and new proposals rely on the assumption that when peers are connected to the network they are always available to answer queries. However, peers may wish to customize the behavior of the system by setting policies for their availability which depend on several factors. For example, the time of the day in which the request arrives, their workload, and the characteristics of the peer that submits the request (it is a member of the ACM group, it is trusted, etc). Moreover, few of such proposals consider the possibility to have a common group ontology for representing the kind of information a group deals with and the possibility that the same concept can have more than one representation (e.g. the author of a book can be represented through the *writer* tag or the *author* tag). Few of such proposals consider that a general agreement between a group member and its coordinator on the meaning of the concepts of the ontology (e.g., a peer may not recognize that the *book author* tag corresponds to the *author* concept in the group ontology) rarely is possible and thus mapping rules are required to map one representation to another. We remark that

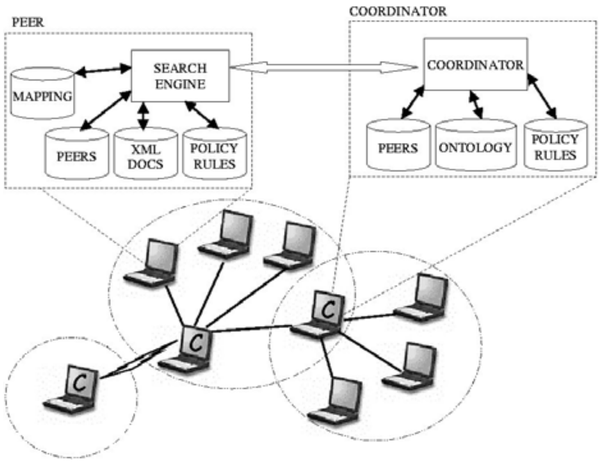

Figure 1. Architecture of our customized semantic-based P2P system

the last two cases can be very common on the Web, where the structure of documents is heterogeneous as well as the background of users delegated to associate the ontology concepts with the structures employed in the peers.

In this paper we present a customizable semantic-based system for the retrieval of XML documents in P2P networks that addresses such missing features of current proposals. The motivating scenario for our system is an environment where entities, such as organizations federated in a "business community" (P. Baglietto, et al., 2002), possess homogeneous sets of documents that are already defined in XML or that can be easily converted into it, and have a strong interest in sharing these documents. The system relies on hybrid P2P architectures (B. Yang and H. Garcia-Molina, 2003) and extends them by performing content-based retrieval of XML documents and enforcing availability policies. The main peculiarity of our system is that all the infrastructure is XML-based. Policies, ontology, mapping rules, and query requests conform to predefined XML schemas. For lack of space, in the paper only the main ideas are illustrated through examples. However, in the companion technical report (G. Guerrini, V. Mascardi, and M. Mesiti, 2005) all the details are provided.

The paper is organized as follows. Section 2 sketches the system architecture and discusses the rationale behind the introduction of ontologies and policies. Section 3 discusses the functionalities offered by the system, namely registration, query evaluation, and document retrieval. Section 4 compares our work with other current research in the field. Section 5 reports concluding remarks and outlines our research agenda.

2. ARCHITECTURE AND MAIN COMPONENTS OF THE SYSTEM

The overall architecture together with the main building-blocks, namely ontologies and availability policies of our system are described in this section. Moreover, we discuss how requests are formulated.

2.1 Peer Organization

The architecture of the system is depicted in Figure 1 and relies on peers and group coordinators.

Peer: Each peer (represented in Figure 1 as a laptop) can dynamically enter and leave any group inside the P2P network. A group represents a set of peers containing information on the same topic. When the peer joins a group for the first time, it is requested to register to the group. After completing the registration stage, the peer can start to submit queries to the system and it can download the system interface that will handle queries from other peers. The interface of each peer is equipped with a local instance of the engine to retrieve documents from the local source and to enforce the availability policies. Any registered peer owns the following data structures:

- The XML documents to share with other peers.

- The set of rules that map concepts from the group ontology to the peer internal representation.
- The set of availability policies.
- The set of peers on the network it is aware of.

This last information is needed to distribute requests arriving to the peer on the network in order to obtain further answers to the queries.

Group coordinator: In a group, a peer plays the role of coordinator (besides continuing to play the role of peer). In addition to the data structures which characterize "common" peers, the group coordinator also owns and manages:

- The group ontology which defines the concepts dealt with by its members.
- The list of all the peers registered to the group, as well as the list of other coordinators which lead groups with related interests.
- The policies of the group, namely those rules that establish the conditions under which the group is available to answer queries.

This information is exploited for coordinating the behavior of the group in answering queries and for minimizing the number of messages the group handles.

2.2 Ontologies

As pointed out from the evaluation of a set of XML documents gathered from the Web (Bertino et al., 2004), different labels can be used for representing the same concept. Such labels can be semantically similar (e.g. *author* and *writer*) or syntactically similar (e.g. *l_name* and *last_name*). Moreover, a label can be used for representing a concept, but the label itself is meaningless (e.g. *T* for time or *P* for paragraph). Therefore, the only use of semantic relationships among concepts of an ontology is not enough to deal with the heterogeneity of XML data on the Web. To face these cases, the ontology is coupled with a set of mapping rules, used for mapping a single concept into multiple representations. In the remainder of this section we present the ontology, the mapping rules, and different approaches for their storage and maintenance in the group. In the description of the ontology we refer to a group of peers, named *CS group*, interested in computer science publications.

Group ontology: The group coordinator owns and manages the group ontology that defines the concepts dealt with by the group members. The group ontology is initially defined by hand by the ontology developer who knows the domain of the group documents. The ontology developer is in charge of maintaining the ontology.

In the ontology each concept c belonging to C (C is the set of concepts) is represented by a pair (w,s), where w is a label and s is the corresponding meaning. The group ontology is organized as a hierarchy specified through a partial relation <=. Given two concepts c_1 and c_2, $c_1 <= c_2$ if c_1 is more specific than c_2. The root of the ontology is the *thing* concept. Moreover, several relationships can be specified between concepts. For example, a *USE* relationship can be specified for representing the fact that a concept is used in the description of another concept.

Mapping rules: The group ontology is coupled with a set of mapping rules. In each mapping rule, a concept is associated with one of the possible syntactic representations adopted by a peer of the group. Formally, a mapping rule is thus a triple $(c, \{p_1, . . . , p_n\}, Q)$, where c (belonging to C) is a concept, $\{p_1, . . . , p_n\}$ is a subset of the group members, and Q is an XSL or XPath expression (W3C, 2004). The meaning of a mapping rule is the following: in $p_1, ..., p_n$ peers the concept c of the group ontology can be mapped to the peer's representation through the Q expression.

Example. Referring to a group ontology containing concepts like document, article, researcher, date, and free_resource, such that article is a specialization of document, the following are examples of mapping rules.

- *(article,* $\{p_1,p_2,p_3\}$*, journal_paper)* denoting that the concept *article* is represented through an element tagged *journal_paper* in the three peers p_1, p_2, p_3.
- *(researcher,* $\{p_1,p_3\}$*, authors/author)* denoting that the concept *researcher* is represented by element *author*, which is nested in the *authors* element, in peers p_1 and p_3.

```
<Policies name = "Claudio">
 <Policy id = "1">
  <TempConstDef name = "TC1">
   <IntervalExpr  name = "SinceFeb1st">
    <Begin> 2/1/05 </Begin>
   </IntervalExpr>
   <PeriodicTimeExpr name = "businessdays"> <StartTimeExpr>
     <Year> all </Year>
     <DaySet>
      <Day> 2 </Day> <Day> 3 </Day> <Day> 4 </Day> <Day> 5 </Day> <Day> 6 </Day>
     </DaySet>
    </StartTimeExpr></PeriodicTimeExpr>
  </TempConstDef>
  <InternalCondition type = "state"  onProp = "PendingRequests"  operation = "LE" value = "15"/>
  <InternalCondition type = "state"  onProp = "CPUIdleTime" operation = "G" value = "50"/>
  <CertCondition onProp = "Institution" operation = "EQ" value = "Department of Computer Science"/>
 </Policy>
</Policies>
```

Figure 2. Example of availability policy

- *(date, {p_1},* <xsl:template match="date"> <xsl:value-of select="month"/>,<xsl:value-of select="year"/> </xsl:template>), denoting that the concept *date* (expressed in the ontology as a string) is represented in the documents of peer p_1 as a structured element containing only the *month* and *year* components.

Alternative organizations of mapping rules: The mapping rules can be maintained in the group coordinator (centralized approach) or distributed among the corresponding peers in the group (decentralized approach). In the last case the group coordinator only maintains the peers that deal with a certain concept. In the remainder of the paper we describe our system following the decentralized organization of mapping rules. We choose this option to avoid the presence of a bottleneck.

2.3 Availability Policies

Current P2P systems assume that, when peers are connected to the network, they are willing to answer queries from any peer, in any time. This assumption however is not always true, as peers might want to restrict their availability to process requests according to the satisfaction of given conditions. In our system, both the group coordinator and the peers can express these conditions by means of policies. Each policy specified by the coordinator expresses group policies and states a common behavior of the group with respect to requests, whereas the policies specified by each single peer represent the availability of the peer, thus they possibly further restrict the group policies. Policies are expressed as a boolean conjunction of conditions that can be categorized in four different kinds:

1. temporal conditions on the time the request is received, either expressed as an absolute or periodic time specification, expressed according to a language inspired by (M. Niezette and J. M. Stevenne, 1992); examples are "not on office hours", "on business days";
2. internal state conditions on the receiving peer; examples are "if there are no more than five requests to be processed", "if CPU idle time is at least 50%";
3. connection conditions on the kind of connection of the peer to the network; an example is "if connected through a LAN";
4. credential conditions (M. Winslett et al., 1997) on the requesting peer; an example is "if the requesting peer belongs to a university department".

Each peer specifies a policy through a conjunction of previous conditions. Different policies can be associated with different periods of time, in which the policies are enabled. For example, a peer might specify that it is available to answer queries during working time (i.e., between 9 a.m. and 17 p.m.) if the workload of the machine does not exceed the 20% of its possibility, whereas, during the rest of the time, it is available to answer queries only from members of the groups it belongs to. When the policy is enabled and the condition holds the request is processed, otherwise it is rejected.

Policies both at coordinator and peer level are represented through XML documents. Figure 2 shows an example of XML document specifying the policy that peer “Claudio” starting from February 1st, 2005 during

```
<Request ID="Claudio:1357" TTL="5" BS="1"  typeofrequest="ANSWER" peer="Claudio">
 <Query> <Article>
  <Researcher operation="EQ">Maria</Researcher>
  <Data operation="EQ">October,2003</Data>
  <Journal operation="EQ">VLDB</Journal>
 </Article> </Query>
 <Credential>
  <Name>Claudio</Name>
        <Nationality>Italy</Nationality>
        <Age>34</Age>
 </Credential>
</Request>
```

Figure 3. Example of request

business days, will answer queries only if they are submitted from a computer science department, the CPU idle time is at least 50%, and there are no more than 15 pending requests.

2.4 Requests

Queries in the requests are expressed using a fragment of the XQuery language (W3C, 2004) corresponding to filtering queries. Such a kind of queries can be easily mapped into a tree representation. The internal structure of the tree represents the structures of the documents possible answers to the query. Moreover, tree leaves contain conditions that content elements of the requested documents should meet.

The tree representation of the query is coupled in a request with:

- An unique identifier. It is generated by concatenating the peer identifier with a unique request identifier in the peer. Peers and coordinators use it to answer/forward at most once the same request.
- A TTL (time to live) counter. It specifies the maximal distance between the sender of the request and its last receiver. This means that, when a group coordinator receives a request with TTL = 0, it does not forward it to other coordinators, but just collects the answers from the peers of its group and returns the documents to the requesting peer.
- A BS (broad search) counter. It specifies the fraction of peers a peer receiving a request can forward to. For example, if BS = 1, the query is forwarded to all the possible peers, whereas if BS = 0.5, the request is forwarded to 50% of possible peers.
- A credential card. It contains properties of the requesting peer and the time of the request. The properties are certified by a Credential Authority (M. Winslett et al., 1997).

An example of request is shown in Figure 3. The corresponding query `article[researcher="Maria" and date="October, 2003" and journal="VLDB"]` looks for articles written by Maria and published in October 2003 on the VLDB journal. The request also contains the submitting peer credential, including values for properties name, age, and nationality.

3. FUNCTIONALITIES OF THE SYSTEM

The functionalities of the system concern with registration of a new peer to the system, routing of queries, and evaluation of queries both at a group level and at a peer level.

Registration: When a client of our system registers to a group, a graphical interface showing the group ontology appears to him/her. The client can browse the ontology and read the textual explanation of each concept. He/she can thus realize which concepts are dealt with by the documents he/she is willing to share (recall that, considering the scenarios where we plan to use our system, we are assuming that a client holds and is willing to share homogeneous documents, and that he/she is aware of their structure), and can insert the syntactic representation of the tags inside his/her XML documents whose semantics is given by some concepts in the ontology. If we consider the motivating scenario of a business community, where the ability to effectively share documents among the participants gives a concrete advantage to the community as a whole, in terms of both saved time and gained productivity, the effort required to peers to register becomes absolutely bearable.

Query routing: When a peer wishes to submit a query in the network, it extracts the peer identifiers from the peer storage. Among them there are the coordinators of the groups it belongs to, but also peers it is simply aware of. The query is submitted with a request of answer to the group coordinators, whereas it is submitted with a request of forward to other peers. We distinguish among these two kinds of requests in order to improve the performance of our architecture and to avoid bottlenecks both in peers and coordinators. Moreover, by considering the request of forward, peers are able to forward a request also to the network borders. When a request of answer is sent from a member of the group to its coordinator, the mapping rules introduced in Section 2 are applied in the inverse mode, in order to obtain a query understandable from the group coordinator. This transformation is performed by the interface of the system local to the peer.

When a query with a request of forward arrives to a peer, the peer forwards the query to its group coordinator, which is in charge of identifying the peers in the group that can contain an answer to the query. The peer can also forward the query to other peers it is aware of, depending on the TTL and BS values. Since a request of forward does not overload a peer, its availability policies are not evaluated. The query is simply forwarded.

When a coordinator receives a query, it applies the policies of the group. Once it has established that the group is willing to consider the request, it determines the members of the group to which the query should be submitted with a request of answer and collects the answers. Moreover, the coordinator can forward the request to other coordinators. When a peer receives a query (with a request of answer) from its coordinator, it first checks whether this is the first time it receives such a request (by considering the ID attribute of the request document) and then applies its local policies for establishing whether it is available to answer the query. Each query answer is coupled with a degree of relevance and a document signature, that is, a number that uniquely represents the content of the answer. The document signature, which relies on DOMHASH (H. Maruyama, et al., 2000), is useful for eliminating duplicates.

When a coordinator receives an answer (either from a coordinator of another group or from a peer inside the group), it checks that the same answer has not been provided yet (through the document signature) and returns it to the requesting peer. The results of a query are thus returned to the requesting peer by passing through the group coordinators which are in charge of eliminating duplicates. The final result that arrives at the requesting peer is thus quite minimal. We remark that the use of the TTL and BS counters coupled with the elimination of duplicates performed by the group coordinators are really relevant in order to reduce the use of the network and to reduce the work of each peer in the network.

Query evaluation in a group: Once the coordinator has established that the group is available to answer the query, it matches the query against the ontology. This match has two purposes. First, the coordinator can identify whether in the group there may be answers to the query. The idea of the match is, indeed, to identify whether the query poses constraints on concepts present in the ontology. Then, since the coordinator knows which peers in the group deal with which concepts (this information is provided by each peer during the registration stage), it can identify the peers in the group containing the concepts appearing in the query. The coordinator then waits for their answer. Each time an answer is returned, the coordinator locally stores its document signature and, in case the answer is new, returns it to the requesting peer. When the coordinator receives the last answer or a pre-fixed elapsed time lasts, the document signatures are removed and the query evaluation process is concluded.

Query evaluation in a peer: When a peer in the group receives a request from its coordinator it first applies its policies in order to determine if it is available to answer the query in that moment. Whenever a peer is available, the query evaluation process starts. In this process, the structural similarity between the query and documents possible answer to the query is considered. Moreover, content conditions (associated with elements of the query) are considered in order to evaluate the content similarity. The structural and content similarities are finally integrated in order to compute the relevance of the document with respect to the query. This process is thus composed of four tasks: 1. query re-writing, 2. evaluation of structural similarity, 3. evaluation of content conditions, and 4. computation of relevance degree. The result of the query evaluation process is a set of documents each one coupled with a relevance degree to the query. This value is then used for ranking the documents to return to the requesting peer.

4. COMPARISON WITH RELATED WORK

In this section, we briefly survey some P2P systems born with the purpose of sharing knowledge (as our system), and we exclude from our analysis those systems developed for sharing computing resources, such as SETI@home and Avaki, and for allowing application-level collaboration between users, such as Groove and Magi P2P technology. We consider seven P2P systems that deeply differ from one another both w.r.t. the motivation and nature of the proposal, and w.r.t. the intended application domain: Edutella (W. Nejdl, et al., 2002, 2003), FreeNet (freenet.sourceforge.net), KEEx (M. Bonifacio, et al., 2004), Napster (www.napster.com), Piazza (A. Halevy, et al., 2004), the Trusted Computing P2P Architecture (R. Sandhu, et al., 2005.), and SWAPSTER (P. Haase et al., 2004, C. Tempich et al., 2004). The comparison considers the two features that more than others characterize our proposal: use of ontologies to answer data requests and to better route them; and use of policies to allow a flexible access to the peer's resources.

In Edutella, each peer defines the supported metadata schemas, and a mapping service translates between different metadata vocabularies; sharing policies are not supported. FreeNet does not include ontologies, but it supports completely anonymous methods for storing and retrieving information. KEEx supports both semantic coordination, ensured by partial and approximate representations of the world, and policies, limited to conditions on the identity credentials of a peer. Napster provides neither ontologies, nor sharing policies, while Piazza, although not integrating ontologies, supports an arbitrary graph of interconnected schemas described in XML. Also, data owners can specify access control policies declaratively and generate data instances that enforce them. The Trusted Computing P2P Architecture only addresses the problem of enforcing customizable sharing policies, that may include conditions on the identity credentials of a peer, on its role, on the platform environment, and on general access rights. Finally, as far as SWAPSTER is concerned, in the application described by P. Haase et al., 2004, each peer has an ontology associated with it, that defines the content of the resources shared by the peer and that refers to an ontology common to all the peers in the system. The application described by C. Tempich et al., 2004, provides security mechanisms, delegation of access rights, etc. Ontologies have been exploited to perform content-based information retrieval in other P2P systems in many ways. For example, S. Castano, et al., 2003, consider a system in which peers are clustered with respect to their interests. Each peer is equipped with an ontology which describes its resources. In the evaluation of a query, a matching manager performs the comparison between a target concept required by the query and the peer ontology, in order to find approximate answers to the query. A "service-oriented" approach is proposed, for example, by D. Elenius and M. Ingmarsson, 2004, that suggest to enhance JXTA with semantic models of services using OWL.

As far as ontologies are concerned, the system we propose differs from the above systems in several aspects. First, in our approach the ontology is associated with the group coordinator which is in charge of determining, by matching a query expressed in XQuery against the ontology, the peers that can contain possible answers. Second, each peer is aware of the groups it belongs to because it explicitly registers to them. This is particularly relevant when a peer wishes to identify the nearest group interested in a particular topic. Third, our mechanism for answering queries relies on mechanisms developed for classifying and filtering documents (E. Bertino, et al., 2004). The query is expressed as a labeled tree representing the pattern of the documents possible answers to the query. The filtering process properly evaluates the structural and content similarity that exists between the pattern and a document in order to return an answer with its relevance degree. This feature combined with the exploitation of ontologies allows us to return more precise answers.

As far as sharing policies are concerned, only the Trusted Computing P2P Architecture provides a flexible declarative language to express complex conditions on the peer's availability; the other systems support the definition of simpler mechanisms to ensure the respect of access rights on the peer's resources.

Finally, as far as routing is concerned, G. Koloniari and E. Pitoura, 2004, present recent results for routing queries among highly distributed XML documents. Their architecture relies on a hierarchical organization of peers connected through a main channel. Our work differs from that of G. Koloniari and E. Pitoura because we consider the presence of an ontology in the group coordinator and mapping rules in each member of the group. In (H. Chalupsky, 2000; A. Doan, et al., 2003; E. Rahm and P.A. Bernstein, 2001) different approaches have been presented for modeling mapping rules. Our mapping rules are simpler than the ones proposed in those papers, but they are expressive enough for the purpose of our system, and – most important - they prevent the issues, described by I. Tatarinov and A. Halevy, 2004, of mapping rule composition.

5. CONCLUDING REMARKS

In this paper we presented a customizable semantic-based system for the retrieval of XML documents in a hybrid P2P network in which peers are organized into groups. The system relies on the use of ontologies and on approximate structural and content matching to determine the most appropriate answers to queries. Moreover, it exploits availability policies that allow peers to customize their participation to the P2P community. All the data structures the system relies on, as well as the routing and query answering algorithms, have been carefully designed and a full prototype based on JXTA, and using the Jena semantic web framework (http://jena.sourceforge.net/) for the ontology management, will be ready soon.

Our current work consists in completing the implementation of the full prototype of our system, in order to verify the impact of our design choices in terms of efficiency, scalability, and accurateness of answers. We also plan to develop an alternative version of the system in which the management of the mapping rules is centralized, in order to verify the effects on the performance of the system.

The current version of our system is designed to support information pull, that is, given a request it is able to identify answers in the network. As a future work we wish to support information push as described by S. Idreos et al., 2004, where queries are permanently posted on super-peers and notifications are sent to the posting peers whenever a document that matches the query constraints is identified. Considering information push in our system requires to enhance the availability policies in order to return notifications and to set when the query should be evaluated. For example, the coordinator can state that a permanent query should be evaluated only during the night.

ACKNOWLEDGEMENTS. We acknowledge P. Bouquet, I. Clarke, E. Franconi, W. Siberski, and S. Staab for their precious advices in drawing the comparison with related work.

REFERENCES

P. Baglietto, et al., 2002. Deployment of Service Oriented Architecture for a Business Community. EDOC, 293-304.
M. Bonifacio, et al., 2004. KEEx: A Peer-to-Peer Solution for Distributed Knowledge Management. Symp. on KM.
E. Bertino, et al., 2004. A Matching Algorithm for Measuring the Structural Similarity between an XML Document and a DTD and its applications. Inf. Systems, 29(1):23-46.
S. Castano, et al., 2003. Ontology-Addressable Contents in P2P Networks. SemPGRID Workshop.
H. Chalupsky, 2000. OntoMorph: A Translation System for Symbolic Knowledge. KR, 471-482.
A. Doan, et al., 2003. Learning to Match Ontologies on the Semantic Web. VLDB J., 12(4):303-319.
D. Elenius and M. Ingmarsson, 2004. Ontology-based Service Discovery in P2P Networks. P2PKM '04.
A. Halevy, et al., 2004. The Piazza Peer Data Management System. IEEE TKDE, 16(7).
P. Haase et al., 2004. Bibster - A Semantics-Based Bibliographic Peer-to-Peer System. Semantic Web Conf., 122-136.
S. Idreos et al., 2004. P2P-DIET: One-Time and Continuous Queries in Super-Peer Networks. EDBT, 851 -853.
S. Gribble, et al., 2001. What Can Database Do for Peer-to-Peer? WebDB, 31-36.
G. Guerrini, V. Mascardi and M. Mesiti, 2005. A Semantic Information Retrieval Advertisement and Policy Based System for a P2P Network, Tech. Rep. DISI, Univ. Genova.
I. Klampanos and J. Jose, 2003. An Architecture for Peer-to-Peer Information Retrieval. SIGIR, 401-402.
G. Koloniari and E. Pitoura, 2004. Content-Based Routing of Path Queries in Peer-to-Peer Systems. EDBT, 29 – 47.
H. Maruyama, et al., 2000. Digest Values for DOM, IBM TokyoRes. Lab.
D. Milojicic, et al. , 2002. Peer-to-Peer Computing. Technical report, HPL-2002-57R1, HP Labs Palo Alto.
W. Nejdl, et al., 2002. Edutella: A P2P networking infrastructure based on RDF. WWW.
W. Nejdl, et al., 2003. Super-Peer-Based Routing and Clustering Strategies for RDF-Based P2P Networks. WWW.
M. Niezette and J. M. Stevenne, 1992. An Efficient Symbolic Representation of Periodic Time, Conf. on I&KE.
E. Rahm and P.A. Bernstein, 2001. A Survey of Approaches to Automatic Schema Matching. VLDB J., 10(4):334-350.
R. Sandhu, et al., 2005. Peer-to-Peer Access Control Architecture Using Trusted Computing Technology. SACMAT.
C. Sartiani, et al., 2004. XPeer: A Self-organizing XML P2P Database System. P2P&DB.
I. Tatarinov and A. Halevy, 2004. Efficient Query Reformulation in Peer Data Management Systems. SIGMOD.
C. Tempich et al., 2004. XAROP: A Midterm Report in Introducing a Decentralized Semantics-Based Knowledge Sharing Application, PAKM Conf., 259-270.
M. Winslett, et al., 1997. Using Digital Credentials on the World Wide Web. Computer Security J., 5(3): 255-266.
W3C, 2004. XQuery 1.0 and XPath 2.0 Formal Semantics.
B. Yang and H. Garcia-Molina, 2003. Designing a Super-peer Network. ICDE.