

## XOP: Sharing XML Data Objects through Peer-to-Peer Networks

Itamar de Rezende, Frank Siqueira  
*Department of Informatics and Statistics*  
*Federal University of Santa Catarina*  
*Florianópolis, Brazil*

*itamarrezende@terra.com.br, frank@inf.ufsc.br*

### Abstract

*This paper describes an architecture for sharing data in XML format through a peer-to-peer (P2P) network, which interconnects devices ranging from high-end servers to mobile devices. The XOP architecture allows devices with low storage capacity and subject to disconnection to use other peers of the P2P network for storing data. Users are able to locate and access data in a friendly and flexible way. Shared data is described using an extensible ontology, built using the OWL language, in order to provide more precise search results. This paper also describes a prototype of this architecture and analyzes the obtained performance.*

### 1. Introduction

One of the main challenges in the future of information technology is the sharing of large amounts of distributed unstructured data.

The World Wide Web, which is based on the client/server model, provides access to information centralized in servers, which can be accessed by clients through their web browsers. This approach requires a high investment on an infrastructure for storing and maintaining data, and allows governments and companies to control information. The Web does not contextualize data in a way that allows for more efficiency in the return of search data. The lack of a native content location mechanism resulted in the creation of search engines, which require a large computing infrastructure, consume a large amount of bandwidth, and are supported by billions of dollars spent on advertisement.

Peer-to-Peer (P2P) networks [1], on the other hand, provide a more dynamic, flexible and adaptive infrastructure than the World Wide Web. P2P networks became popular among the Internet user community

due to the transfer of audio and video files. Popular file sharing applications such as Napster, Gnutella and Kazaa have attracted attention to P2P technology. However, this technology may be employed in other areas, such as information sharing, location and retrieval.

In order to provide a more efficient way for sharing data on the Internet, a new architecture called XOP (XML data Objects in Peer-to-peer networks) has been proposed by the authors of this paper. This infrastructure allows the storage, sharing, location and retrieval of information over a partially centralized P2P networking infrastructure [12]. The XOP architecture allows the sharing of information available in a distributed environment, comprising of fixed and mobile devices interconnected through a P2P network. Data semantics is described using an ontology, which allows shared content, stored in XML [3], to be bound to a meaning. A query interface allows end-users to search for information distributed through the P2P network in a friendly and intuitive manner. In addition, mobile devices, which are subject to disconnection, are allowed to store content on other peers, allowing the access to this information even if they are offline.

The proposed architecture was designed with the aim of providing a precise, easy to use information storage and retrieval service, hiding from the user the complexity brought by the distributed nature of the environment. Additionally, the architecture was developed with the intent of allowing the integration of information distributed across heterogeneous devices, taking into account the limited processing, communication and storage resources provided by mobile devices.

The XOP architecture may be used in different scenarios, such as: a passenger searching for timetables and routes of public transportation; a tourist searching for a hotel in a specific location; a company trying to obtain the best quotation for a product among several suppliers; and many others. Peers may range from mobile phones and PDAs to large data centers.

The structure of this paper is organized as follows: section 2 presents the state of the art in the area of information sharing and retrieval in P2P networks; section 3 provides an overview of the XOP architecture, describes how it works and analyses its features in face of other proposals found in the literature; section 4 describes the prototype implementation of the architecture and evaluates its performance; and finally, section 5 presents the conclusions and suggestions for further development of this work.

## 2. Sharing Data in P2P Networks

Despite their current popularity, commercial P2P applications allow access only to content encoded using data formats that provide metadata for content indexing – e.g. audio and video files. These can be easily indexed using one (or multiple copies of a) centralized server, or through a set of specialized peers which maintain a distributed hash table (DHT). Both strategies rely on metadata (e.g., song name, artist, encoding format) instead of indexing the content itself.

Sharing, locating and retrieving information is more complicated than doing the same based on metadata associated to audio and video files. In this case, indexing metadata might not be sufficient for providing good search capabilities. Due to this limitation, the use of P2P networks for sharing, locating and retrieving information is still in its early development stages [7]. A few proposals are found in the literature, but none of them have reached the status of a commercial solution.

### 2.1. Related Work

Bricks [9] is an example of P2P infrastructure for sharing information based on metadata. This project is aimed at developing and maintaining an infra-structure that allows for the sharing of cultural knowledge, such as digital libraries, on a partially centralized P2P network. Each node in the network represents a member institution, and runs the software that provides access to the Bricks network. Some of these nodes, called BNodes, store content and manage metadata. Each BNode only recognizes a subset of BNodes. Thus, if a BNode wants to reach other unknown members, it must request a known member to propagate the request. An XML database, maintained by the BNodes, allows the association of metadata to shared documents, which is then stored in a distributed hash table (DHT). The documents are divided in small parts and are distributed for storage within the

community. Documents are created and maintained by the community and can be accessed by any peer, with no concern for data location. Data access is done using an infrastructure based on web services technology.

DBGlobe [8] is a P2P global management system that allows users to describe, maintain and consult data. DBGlobe allows the connection of anonymous and mobile devices, called Primary Mobile Objects (PMOs), which store information on a context and connect to a Cell Administration Server (CAS), which provides connectivity and other basic services to PMOs, so that they can publish and discover information. DBGlobe creates a partially centralized P2P network and uses ontologies for the description of data shared in the network. PMOs, according to their capacity, can be used for sharing data, moving data around, updating data or simply transferring it. Information is exchanged between PMOs, which act as web services. The available services use the Active XML paradigm to invoke and execute services and exchange data. However, DBGlobe does not provide mechanisms for handling the disconnection of mobile peers, and search mechanisms do not deal with data semantics.

Piazza [4] provides a P2P infrastructure for building Semantic Web applications. Piazza offers a language for mediating between data sources on the Semantic Web, which maps both domain and document structure. A Piazza application consists of many nodes, each of which can supply data which is mapped to a schema, provide only a schema (or ontology), or both. A very simple node might only supply data (perhaps from a relational database); at the other end, a node might simply provide a schema or ontology to which other nodes' schemas may be mapped. Piazza also enables interoperation of XML data with RDF data associated to rich OWL ontologies. Mappings in Piazza are provided locally among small sets of nodes, and the query processing algorithm is able to locate and obtain relevant data distributed across the Piazza network.

KEEx [2] is a P2P system which allows a collection of Knowledge Nodes (KN) to search and provide documents on a semantic basis without requiring a previous arrangement on how documents should be classified, or on a common language for representing semantic information within the system. In KEEx, peers are organized in knowledge communities, and the following principles are implemented in a quite straightforward way: (i) each peer provides all the services needed by a KN to create and organize its own local knowledge (autonomy), and (ii) by defining social structures and protocols of meaning negotiation in order to achieve semantic coordination (e.g., when searching documents from other peers). KEEx uses a

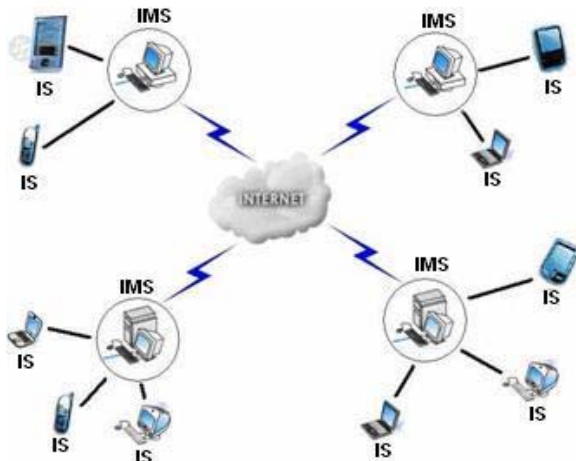


Figure 1. The XOP Architecture

semantic mapping across autonomously developed schemas for organizing and retrieving available data (e.g., classification schemas, database schemas, directory structures). Both KEE<sub>x</sub> and Piazza do not provide any kind of support aimed at mobile peers.

### 3. The XOP Architecture

The XOP architecture provides storage and search mechanisms that can be used in a partially centralized peer-to-peer (P2P) network. In order to allow the description of shared content with semantically meaningful information, a standard way for formalizing content and concepts related to a domain is defined by the architecture. In addition, the architecture provides facilities aimed at mobile peers, which are allowed to share, search and retrieve data despite being subject to disconnection from the network.

#### 3.1. Overview of the Architecture

The XOP architecture comprises a set of supernodes, which are responsible for acting as interconnection hubs for regular, anonymous peers. Supernodes are expected to have higher processing and storage capacity, and must also have a reliable connection to the network. Supernodes execute the Information Management Service (IMS), which interacts with the Information Service (IS) running on every peer connected to the network, as illustrated by Figure 1.

The IMS, as shown by figure 2, is formed by services responsible for managing information in the network, receiving requests from peers, storing data in the XML Database, and propagating queries to other IMS's. These services are:

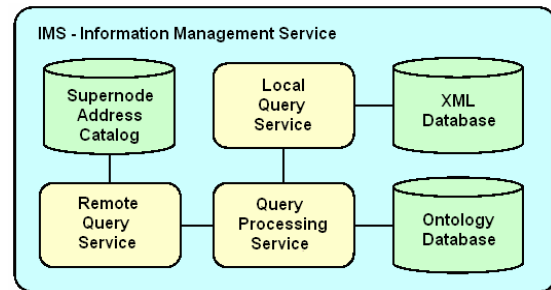
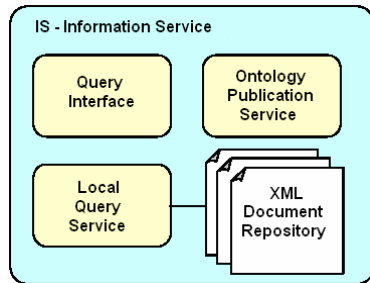


Figure 2: Information Management Service

- Query Processing Service (QPS);
- Local Query Service (LQS);
- XML Database (XML-DB);
- Ontology Database (ODB).
- Remote Query Service (RQS);
- Supernode Address Catalog (SAC);

The IMS connects to the P2P network on startup and then initializes the **Query Processing Service (QPS)**, which is responsible for receiving requests issued by peers connected to this supernode. The **Local Query Service (LQS)** is responsible for locating content on the local **XML Database (XML-DB)**. The XML-DB is the database where the contents shared by peers connected to this supernode that are unable to store the content or to maintain a connection with the network. The **Ontology Database (ODB)** is an XML database that contains the semantic description of content shared by peers connected to this supernode. This information is employed by the LQS for providing more accurate responses to queries issued by users. The **Remote Query Service (RQS)** propagates the query to other supernodes connected to the P2P network. The addresses of other supernodes are kept in the **Supernode Address Catalog (SAC)**.

The IS, on the other hand, is a service running on network peers – i.e., any device with connective capacity that queries and/or makes information available through the network. Differently from supernodes, regular peers are not required to have high processing and storage capacity, i.e., may be PDAs or mobile phones connected to the network to access shared data, with no data storage function, or may be also desktop computers sharing information through the network. A regular peer is not required to have a permanent connection to the network – it can be a mobile device that connects to the network only for issuing a single query. Peer “A” only communicates with peer “B” after receiving its address from the supernode, and “B” can be connected to a different supernode. Regular peers can choose to serve as an information storage site, if it has the necessary storage



**Figure 3. Information Service**

capacity and wants to have more control over the information it shares on the network; or it can store it directly in the supernode to which it is connected. Peers can connect to the network at any time, and need no special permission as long as the network has got the service locally installed.

The Information Service (IS), as shown by figure 3, is formed by the following components:

- Query Interface (QI);
- Local Query Service (LQS);
- Ontology Publication Service (OPS);
- XML Document Repository (XDR).

The **Query Interface** (QI) is the application responsible for allowing the user to create and execute the query and then verify the obtained results. This application leads the user in a friendly manner through the steps required for obtaining the result of the query. The user must initially define, through the QI, which characteristics identify the content he wishes to locate. As soon as these characteristics have been defined, the query is submitted to the IMS running on the supernode which is connected to this peer. The supernode processes the query and returns a list of contents that fulfill the characteristics requested by the user. This list contains the complete content characteristics and corresponding network addresses, which can report to any other peer (including supernodes) connected to the network. The user can then select a content to know details of its structure. Up to this point, the user has not specified any constraint on the located data, but has only identified characteristics of the content he wishes to locate. The only connection necessary from the issuing peer is with its supernode, which is responsible for knowing and contacting the other supernodes. The regular peers are not involved in this first part of the query processing. Once the content has been defined, it is possible to specify constraints on the data to be located. The query result is returned to the issuing peer and the user can then verify the results and even create a local copy of the obtained data.

The IS can store XML documents locally if it has storage capacity, or may send them to be stored directly in its supernode. In the first case, documents will be stored in the local **XML Document Repository (XDR)**. Any shared content must be semantically described through the **Ontology Publication Service (OPS)**. This process may be as simple as binding the shared content to concepts already described by other users, or on extending the ontology if an appropriate description for the shared content is not already available.

If the peer chooses to store content locally, the **Local Query Service (LQS)** will be responsible for consulting the contents requested by other peers in the XDR. The LQS receives a query in XML format through the network and searches among its local XML documents for those that fulfill the request. The results found are sent to the requesting peer.

A reduced version of the IS, containing only the Query Interface, is made available to be executed on devices that wish only to issue queries on the network without sharing data.

### 3.2. Data Sharing

Making the information available in an organized way, in which content returned to the user is clear and coherent with the specified search parameters, is a quite complex process. In order to provide more accurate search mechanisms, the XOP architecture employs an ontology for content description, which can be further extended to describe novel concepts. The use of an ontology aims at organizing concepts in such a way that any content can be published or accessed. This approach contrasts with the limited way in which searches are performed in current P2P networks that usually adopt a search procedure based on metadata associated to the content, which is indexed using distributed hash tables (DHT). The ontology provided by the XOP architecture is based on the standard recommended by the W3C – the Ontology Web Language (OWL) [11].

In order to build such data sharing environment, the first step is to provide a means for users to share locally stored content. This is achieved through the provision of a tool that allows the user to share local data, structured in XML format, and describe the semantics of this data using the ontology provided by the architecture. Concepts already described in the architecture may be just referenced by the user, binding his data to the already described concept. New concepts must be described extending the ontology.

The **Ontology Publication Service (OPS)**, available in the IS, allows the user to describe the content it shares on the network. The ontology is based

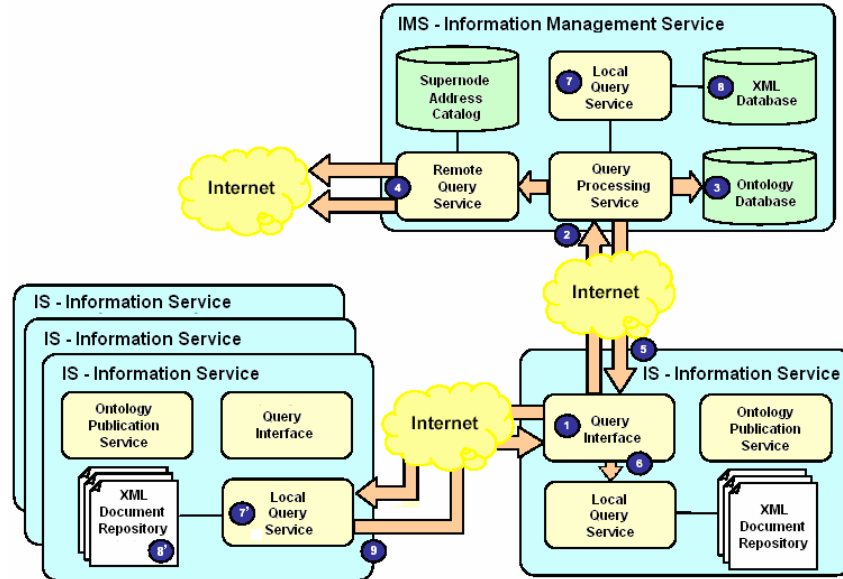


Figure 4. Query Processing

on OWL [11], which describes contents as objects, and characteristics as properties of these objects. The information required to create the ontology is:

- **Class:** identifies a common set of characteristics of a content, equivalent to the concept of object in object-oriented systems. It is a common interest domain. It is composed by:
  - **Name:** a noun identifying this class.
  - **Description:** a text describing this class.
  - **Properties:** describe characteristics of this class. Composed by:
    - **Name:** a noun identifying this property;
    - **Type:** an *ObjectProperty*, when the property has the role of relating this class to another class; or a *DataType*, which defines the type of data (e.g., character, floating point, date, etc.) which corresponds to this property.
    - **Cardinality:** identifies the amount of values assumed by this property. Accepted values are *single* (for just one value) or *multiple* (for more than one value).
  - **Dictionary:** creates associations between words that can be used as synonyms in this context, in order to allow the easier location of this object.

After the ontology that describes content is created, it is sent for storage in the **Ontology Database (ODB)**, located in the IMS. The ODB is a database responsible for recording OWL documents describing shared data. It is important to notice that this database does not contain the data itself; its goal is to be a repository in which the structures that identify shared content and determine the location of the data are stored – i.e., it is basically a data dictionary. Each IMS is responsible for

maintaining in its ODB the description of data shared by the IS's that are directly connected to it. Once stored in the ODB, this ontology is available for other users to extend its structure and to share data with the same semantic structure. Therefore, once the ontology describing data with some associated semantics is created, other peers can share semantically equivalent data without having to extend the ontology – i.e., they just have to specify which ontology describes the data they are sharing.

When new data is shared by a peer, the IMS validates the data based on the specified OWL description. The shared data can be stored in the IS, in an XML document, or in the XML database (XML-DB) of the IMS, if the IS runs on a mobile device with limited storage capacity or subject to frequent disconnection from the network.

### 3.3. Query Processing

At the time of query processing, since the information is distributed in the peer-to-peer network, users have no idea of where the information is located, but are able to identify some constraints that might help to find the data. Therefore, the search process is executed in two phases, as depicted in Figure 4.

In the first phase, the user identifies the semantics of the content he wishes to locate, with no concern for the data itself. At this moment, the user should try to find contents that have semantic characteristics that fulfill the query, typing in the QI (step 1, figure 4), parameters that identify part of the content he is searching for. The query is sent to the QPS (step 2), which searches the ontology database (step 3). The

IMS propagates the search to other IMS's through the RQS (4). For each issued query, the RQS waits for a certain time for the results, at the end of which any response will be ignored. This is necessary to avoid waiting for a return indefinitely. The requesting IS receives a list with either supernodes or regular peers in which content matching the specified characteristics is stored, as well as the content structure (5). In this first phase, the search was only processed by supernodes, without access to regular peers, since the goal of this phase is to identify content matching semantically the information requested by the user.

In the second phase of the query execution, the user specifies filters regarding the data he wishes to locate (6) and submits the query to one or more peers identified in the first phase of the query execution. The IS running on each peer selected by the user receives the query in the LQS (7) and tries to locate XML documents (8) that match the request. In case any IMS has answered as having the desired content, its LQS (7') will also receive the query and will run a search in the XML-DB database (8'). Those that fulfill the constraints send a positive answer to the requesting IS (9), which in turn presents the search results as a list to the user, allowing him to browse through the results and, whenever required, to save the content locally.

Due to the dynamic nature of the P2P network, the originating peer may disconnect during the search, and therefore the results must be stored by the IMS, waiting for the next connection of this peer. Thus, every time a peer reconnects to the network, the service must verify whether there are pending requests. These results are stored by the IMS for some time and are then excluded.

### 3.4. Analysis

The XOP architecture has a number of advantages when compared to other proposals found in the literature and described in section 2.1.

The Bricks project [9] and the XOP architecture have similar characteristics, but while in the Bricks project documents are distributed among peers, in the XOP architecture the document remains located in a single peer. Additionally, the XOP project uses ontologies to describe data, providing greater indexing and data searching facility than the metadata-based strategy used in the Bricks project.

DBGlobe [4], KEEEx [2] and Piazza [4] have similar characteristics to the XOP architecture in regard to the description of shared content using ontologies. However, these architectures are not targeted at mobile devices, so they do not provide mechanisms for dealing with frequent peer disconnection, which may result in unavailability of shared data.

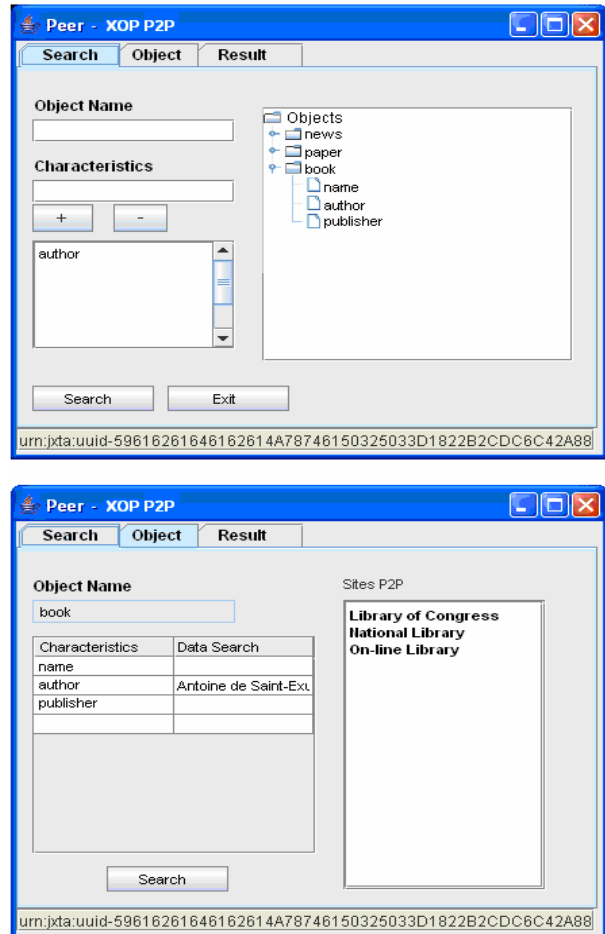


Figure 5. Query Execution in the Query Interface

## 4. Implementation and Tests

A prototype of the XOP architecture has been implemented in Java 5.0. The P2P network was built using JXTA 2.3 [5]. The XML data is processed using SAX 2.0 [10].

Since one of the goals of this work is to create an independent structure that could be available in different kinds of device, instead of using an existing XML database, a new content storage structure for XML documents was developed from scratch.

Figure 5 presents an example of query executed through the Query Interface. In the first part, the characteristic “author” of the required data is informed. After pressing the “Search” button, the IS communicates with the IMS, which replies with three content domains that have the characteristic searched for. The user can select one of the listed contents – in figure 5, the example is “book” – and in the following panel, called “Object”, all the content characteristics are presented. The user informs a value for a

characteristic – in this case, the author name – and submits the search to all the peers where “book” is stored. Peers sharing data that match the search parameters reply by returning the corresponding XML document. In figure 5, three peers, identified as “Library of Congress”, “National Library” and “On-line Library” replied to the search. The user can then select one of the listed peers to obtain the shared data.

#### 4.1. Performance Evaluation

A network with two supernodes and three peers connected to each supernode was built to execute performance measurements. Tests were performed on computers with Intel Celeron 2,80 GHz processor, 512 MB of RAM, running the Windows XP Professional operating system, and interconnected by a 100Mbps Ethernet network.

During these tests, the size of the database of each supernode was raised in order to evaluate the scalability of the architecture. Figure 6 shows the average response times for the two phases of the executed queries – first, searching the ontology database in the IMS; and then, obtaining data from the IS where matching data was found. The graph shows that the architecture scales well even when the database in the IMS is quite large. It is important to notice that the objects stored in the IMS describe different content, so it is unlikely to find so many objects in the database of a single IMS.

#### 5. Conclusion

The XOP architecture, which was described in this paper, allows the description, sharing and discovery of content shared by users. In order to achieve this, data is structured in XML documents and described through an ontology, which represents and elaborates the structure and standardizes the format adopted by shared content. A P2P networking infrastructure is built for sharing data among users. In this environment, any device with a network connection – even a low-connectivity low-processing capacity device such as a PDA or a mobile phone – may generate, store, share and query content.

As suggestions for future work, we would like to highlight: the search for mechanisms to improve query performance; investigate the use of more efficient storage mechanisms; a study on the response time of large-scale queries for evaluating the system response when overloaded; and the implementation of the IS using JXME – an implementation of JXTA for mobile devices with support for Java ME [6].

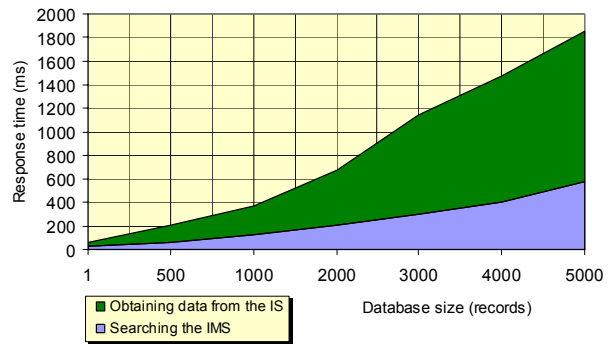


Figure 6. Total Query Response Time

#### References

- [1] Androutsellis-Theotokis, S.; Spinellis, D. *A Survey of Peer-to-Peer Content Distribution Technologies*. ACM Computing Surveys, Vol. 36, 335–371, 2004.
- [2] Bonifacio, M.; Bouquet, P. Danieli, A. Dona, A.; Mameli, G. Nori, M. *KEEx: A Peer-to-Peer Solution for Distributed Knowledge Management*. Proceedings of I-KNOW '04, Graz, Austria, 2004.
- [3] Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler, E.; Yergeau, F. *Extensible Markup Language (XML) 1.0 (Fourth Edition)*, <http://www.w3.org/TR/REC-xml>. Accessed on: January 2007.
- [4] Halevy, Alon Y., Ives, Zachary G., Mork, Peter, Tatarinov, Igor. *Piazza: Data Management Infrastructure for Semantic Web Applications*. WWW2003, May 2003.
- [5] *JXTA v2.3.x: Java Programmer's Guide*. Available at: <http://www.jxta.org>, 2005.
- [6] *JXME – JXTA Java Micro Edition Project*. Available at: <http://jxme.jxta.org>. Accessed on: March 2007.
- [7] Koloniari, G.; Pitoura E. *Peer-to-Peer Management of XML Data: Issues and Research Challenges*. SigMod Record, Vol. 34, p. 6 – 17, 2005.
- [8] Pitoura, E., Abiteboul, S., Pfooser, D., Samaras, G.; Vazirgi-annis G. *DBGlobe: A Service-oriented P2P System for Global Computing*. SigMod Record, Vol. 32, p. 77-82, 2003.
- [9] Risse, T., Knežević, P., Meghini, C., Hecht, R.; Basile, F. *The Bricks Infrastructure: An Overview*, 2004.
- [10] *SAX*. Available at: <http://www.saxproject.org>. Accessed on: January 2007.
- [11] Smith, M. K., Welty, C., McGuinness, D. L. *OWL Web Ontology Language Guide*, Available at: <http://www.w3.org/TR/owl-guide>. Accessed on: January 2007.
- [12] Sung, L. G. A., Ahmed, N., Blanco, R. *A Survey of Data Management in Peer-to-Peer Systems*. Web Data Management, 1-50, 2005.