Managing an XML warehouse in a P2P context*

Serge Abiteboul**

INRIA-Futurs, LRI and Xyleme
Serge.Abiteboul @inria.fr

Abstract. The paper is about warehousing XML content in a P2P environment. The role of an XML warehouse is to offer a centralized entry point to access a variety of information in an enterprise and provide functionalities to acquire, enrich, monitor and maintain this information. We consider a number of reasons to use a P2P approach to such warehouses, e.g., reducing the cost or sharing the management of information. We mention research issues that are raised.

1 Introduction

Enterprises and more generally communities centered around some common interest may benefit from the construction, enrichment, monitoring and maintenance of large repositories of information with methods to access, analyze and annotate this information. We propose to use XML and Web services as the basis for such a warehouse. A warehouse is typically a server that centralizes and manages all the relevant information. In this paper, we consider as an alternative the management of such a warehouse in a Peer to Peer (P2P) environment.

Let us first see the main aspects of these two concepts:

Warehouse [24] The goal of a warehouse is to provide an integrated access to heterogeneous, autonomous, distributed sources of information. The information is acquired in advance, transformed, filtered, cleaned, and integrated. Queries are then evaluated without access to the original sources.

Peer to Peer There are many possible definitions of P2P systems. We mean here that a large and varying number of computers cooperate to solve particular tasks (here warehousing tasks) without any centralized authority. In other words, the challenge is to build an efficient, robust, scalable system based on (typically) inexpensive, unreliable, computers distributed on a wide area network. This leads in particular to complex communication and information discovery schemes.

The combination of the two may seem a bit confusing, so let us try to articulate it more precisely. The data sources are heterogeneous, (very) autonomous

* The research was partly funded by EC Project DBGlobe

** Gemo Team, PCRI Saclay, a joint lab between CNRS, Ecole Polytechnique, INRIA and Université Paris-Sud

and distributed. The warehouse presents a unique entry point to this information. Thus, the warehouse is logically homogeneous and centralized; and a P2P XML warehouse should logically not be distinguishable from a centralized one - to some extent, it is *only* an implementation issue. Now, the physical resources of the P2P warehouse consist of distributed peers. So, the physical organization also consists of heterogeneous, autonomous, and distributed machines, although a higher level of trust may possibly be achieved in this setting.

It is important to keep in mind that the warehouse is *logically centralized* with all peers acting as entry-points and *physically distributed*.

To see an example, consider the management of a very large collection of preprints. Any registered user is allowed to publish preprints. Other preprints may be found, say by a Web crawler. The warehouse is implemented over several peers that store replicas of the preprints. It may be the case that a site (say INRIA) is a source of preprints and is, as well, one of the peers in the warehouse. These two roles should not be confused. In principle, an INRIA preprint may be replicated, from the warehouse viewpoint, at several peers and possibly not even be on the INRIA peer.

The paper is organized as follows. In Section 2, we discuss the functionalities of the warehouse. In Section 3, we consider advantages of implementing it in a P2P environment. In Section 4, we mention research issues that are raised. In place of a conclusion, we briefly present Active XML as a candidate infrastructure for such a warehouse.

2 XML warehouse

In this section, we present the main functionalities of an XML warehouse. We will consider why to implement it in a P2P system and issues that this raises in the following sections.

We are concerned here with a *warehousing of content* to provide a unified view over distributed, autonomous and heterogeneous information sources.

The focus is on warehousing. One typically distinguishes between a mediation approach (the integration remains virtual) and a warehouse approach (the data is materialized and possibly transformed). Both approaches present advantages. In a warehousing approach, we fetch information ahead of time from information sources. In contrast to a mediation approach, the information may be cleaned, enriched, integrated before the arrival of any query.

The focus is on a warehouse of "content" ("qualitative information") rather than typical warehouses that are more concerned with "quantitative information" typically organized in relations. Content has less structure; one sometimes say it is semi-structured. This motivates the use of XML. With XML, one can capture plain text, as well as text with some structure, all the way to very structured information such as tuples. Clearly, the management of meta-data is here an essential component, as advocated in Tim Berners-Lee's view of the Semantic Web [9].

Let us make more precise this notion of content.

Content: physical and logical For us, content is information in its most general form. The primary unit of information is the *content element*. Content elements may be documents, fragments of documents, files, relational tuples, and similar information. We will use XML, the standard for data exchange, as a core data model. Content elements will be transformed, when possible, into XML so that their structure becomes available. Some may stay in their original formats, e.g., pdf files. However, meta information about pieces of information will always be in XML. A content element is a physical notion, but also a logical notion. Typically, a content element may be a portion of a document in its XML incarnation, such as the Subject field of an email, or a collection of such multiple content elements, such as an email folder.

Another essential component of the warehouse is the *global schema* of the warehouse, i.e., the organization that a user would see, for instance, while browsing the warehouse. This aspect will not be considered here.

The main functionalities of such a warehouse are as follows:

Acquisition A goal is to locate content elements in source information systems and load them in the warehouse (pull mode). This is typically based on Web services, Web crawlers or general LTE tools[3] to obtain data from virtually any system, e.g., LDAP, newsgroup, emails, file systems, database systems, etc.

Content may also be acquired from users in *push* mode. This may involve document or meta-data editing (achieved, e.g., via the WebDAV protocol) or explicit publication in the warehouse.

Enrichment The goal is to enrich the warehouse or, in other words, add value to its content. This may involve for instance translation to XML (XML-izer of various forms), structural transformations (e.g., via XSL/T processor), classification, summarization, concept tagging, etc. Enrichment may act at different levels:

1. enriching the meta-data, e.g., document classification;
2. enrichments inside the document, e.g., extraction and tagging of concepts inside the document; and
3. enriching the relationships between documents, e.g., creation of tables of content.

Repository The warehouse should be able to store massive amounts of XML, up to terabytes. It should also support indexing (of data and meta-data) and processing of queries over XML data. Finally, it should provide standard repository functionalities such as recovery and access control.

Change control It is often the case, that users are particularly interested in changes (e.g., a new release of software or the newly published newspaper articles on a particular topic). This leads to issues in versioning and the support of continuous queries [11]. Of particular importance is the monitoring of the loading of information in the warehouse.

[3] LTE stands for load, transform and extract.

View and Integration The system should provides tools for building user views, e.g., by restructuring documents via XSL/T transformations and integrating collections of heterogeneous XML documents. One should also consider proposing automatic or semi-automatic tools to analyze a set of XML schemas and, using some ontologies of the domain, integrate them and then support queries on the integration view.

Exploitation The exploitation of the warehouse primarily involves functionalities such as: querying, browsing, annotating content, generation of complex parameterized reports, and on-line analysis of content.

Administration The warehouse has to be administered. This includes in particular the means to register acquisition and enrichment tasks, the management of users and of their access privileges, as well as the control on-going tasks such as backup and failure recovery.

The functionalities of the warehouse should be available via GUI or via programs. When we move to a Web context, HTTP and Web services are clearly the natural candidates for the interface.

The advantages of such a warehouse approach are discussed in more details in [5].

3 Why a distributed warehouse?

In many contexts (e.g., in an enterprise), the best physical architecture for such a warehouse is a centralized server, or a cluster of machines as proposed by Xyleme [26]. In others (e.g., in a large consortium), a P2P organization may be preferable, e.g., to share the cost of the warehouse as well as its management. This is what we consider here. A main difference is that there is no (or less) centralized authority. In contrast with the machines involved in a cluster implementing an XML warehouse, the machines involved in a P2P warehouse are imposed little constraints; they are typically heterogeneous, less reliable and connected via a much slower network.

Although the architecture is very different, observe that the functionalities remain the same. However, because of the P2P aspect, they take a slightly different flavor. For instance, discovery of data and services becomes a central issue, whereas it was not really an issue in the centralized case (except for the feeding part). For instance, consider a P2P warehouse of pre-prints involving thousands of actors. Finding a particular preprint may require more work since we cannot rely on a single trusted peer to maintain it.

As mentioned in the introduction, this may seem a bit confusing since one typically think of a warehouse as centralized. We would like to stress again the fact that the warehouse is *logically centralized* with all peers acting as entry-points and *physically distributed*. It is also important to see the difference with a mediation approach. In a P2P approach, the content is copied to the warehouse and pre-processed (enriched).

Since the data in a P2P warehouse is distributed between the peers, query processing involves distributed computation (as in the mediation setting). It

is, in general, natural to consider hybrid approaches merging warehousing and mediation. Indeed, it is even much more natural to think in these terms in the P2P setting since both now involve distributed computations. The following table tries to picture the various possibilities:

Integrator is:	Centralized server	Cluster of machines	Machines in P2P
data in integration system	warehouse	Xyleme warehouse	P2P warehouse
Hybrid			
data in source systems	mediation		P2P mediation

In the following of the section, we consider advantages of P2P warehouses. There are first the standard advantages of distribution:

- Performance. Better performance may be achieved by avoiding the bottleneck of a centralized server and using data replication and caching.
- Ownership. Each peer may keep full control over its own information. This does not have to be the rule but in some cases, this may allow to let the responsibility of access control to the owner of information.
- Cost. This avoids the cost of a centralized server and enables taking also advantage of local unused resources (storage and processing). Also, with this approach the costs may be shared between the peers, e.g., the peers that participate in a topic-driven crawl of the Web may share the network cost.
- Dynamicity. Peers may enter and leave the system in a transparent manner whereas it is typically difficult to add/remove new sources of data in a centralized setting.

These advantages come at some cost:

- Performance. Complex queries over distributed collections may get real expensive, in particular, communication cost may become high.
- Consistency maintenance. It is much more complicated to keep data consistent in a P2P setting. In particular, this is not the right framework for OLTP applications.
- Quality. Quality may be more easily controlled in a centralized setting. However, for some aspects of quality such as reliability and availability, the distributed approach has very good answers based on replication.
- Complexity. There is the extra complexity of managing distribution, e.g., replication of documents.

Example 1. Distributing a warehouse of preprints. Each peer contains some preprints. Each peer knows about other peers that are part of the preprint P2P warehouse. They do not need to use the same physical organization. One can query any peer to find information about preprints. Peers know how to route queries to the appropriate peers. All are willing to use some common tools to make this work. Installation and linking of these tools should be 0-effort.

Based on the previous discussion, one may try to sketch the main lines of a typology of P2P warehouses vs. centralized ones:

- decentralized organizations: the P2P approach may be more appropriate for a loose consortium, in absence of any centralized authority and centralized accounting.
- lower end applications: the P2P approach allows developing warehouse applications at zero cost when information quality and access control is not critical.
- upper end applications: the P2P approach may be appropriate to provide better performance, more availability and reliability.

Observe that the two approached may be combined. One company may, for instance, decide to use "content marts" implemented on clusters of machines in local area networks. These marts are then made to cooperate in a global P2P environment. This is in the spirit of grid of clusters.

4 Some issues

From a technical viewpoint, the main issue is *distributed data management*, by no means a novel issue [22]. However, in a P2P environment, the context and in particular, the absence of central authority and the number of peers (possibly thousands to possibly millions), change the rules of the game. Typically, problems that have long been studied take a different flavor here.

In this section, we illustrate the problem with some issues that are raised. This is certainly not meant as a comprehensive survey of all the research issues in the area. It is voluntarily biased by the interests and recent works of the author.

Information and service discovery This is the most classical technical problem and research has already been considering it in the global setting of the Web, see, e.g., Object Globe [10]. An important feature is that communications may be slow or blocking and query processors should be built with that in mind.

The core problem is perhaps that of "look-up". Given a piece of information specified by a key or just some attribute value, find the relevant information in a P2P system over a changing set of nodes. Starting from centralized techniques such as the one used by Napster, there have been a lot of work in this area, in particular around dynamic hash tables, see, e.g., [8, 17].

Note that such indexing may be used to discover data, but also services of interest. This is related to emerging world-wide efforts to publish services, e.g.,

UDDI. UDDI is promoting a centralized repository of Web services. It would be interesting to consider systems that would provide the same functionalities in a P2P setting.

Web crawling and page ranking Web crawlers such as Google crawl the Web and rank pages according to certain criteria. A community may want to organize its own crawl to adapt it to its needs: (i) be able to crawl private portions of the Web or (ii) guide discovery and refresh of pages based on the precise interest of the community.

Some Web crawlers are implemented on clusters of PCs (e.g., Google or Xyleme). In some sense, the cooperation between different peers is similar to the cooperation between the PCs in a cluster with different Web sites a priori distributed between the various machines. Communications are then needed between peers to send to other peers the list of discovered URLs they are responsible for. The computation of page rank in this setting is more challenging. We are investigating this issue starting from an algorithm that does not require storing the graph of the Web [6].

P2P mediation In a centralized setting, one usually assumes that the semantics of the integration is known in advance, e.g., by some mapping rules from the different sources to a global schema or via ontologies that describe the correspondences. See, e.g., [19] for a survey on semantic integration. In some communities, such well accepted ontology may exist. However, in many cases, there may be several related ontologies with bridges between them that should be used to derive the mappings between the sources. This challenging problem is, for instance, the topic of [16, 14, 18].

Monitoring We developed a monitoring system for the Web based on a centralized warehouse [21]. It would be interesting to "move" such functionalities to a P2P setting. This will best illustrate various aspects of P2P warehousing. So, more precisely, consider a very large collection of documents and a very large number of users who want to be notified when some particular documents they are interested in change. For sources that are willing to participate in this monitoring (willing to be peers in the system), it definitely makes sense to let them monitor their own information. Other information sources may not be able to support such monitoring or may not be willing to participate in the P2P system. Then, the various peers have to share the task of monitoring the information of such sources, thereby avoiding the bottleneck of a centralized server and possibly taking advantage of "network locality".

5 Advertisement: Active XML

We believe that peer to peer data management will become more and more popular. Research projects are already addressing some aspects of the problem, e.g., Piazza [23, 15] at U. Washington, PlanetP [12] at Rutgers U. Active XML

[2] at INRIA, P-grid [1] at EPFL Lausanne or the P2P project at ETH Zurich [13].

We next consider Active XML (AXML for short), a language and a system under construction at INRIA as the basis for developing such a P2P warehouse. A more detailed description of AXML may be found in [20] whereas some essential extensions of AXML to support the distribution and replication of portions of a document may be found in [3].

An Active XML document is an XML document where some of the data is given explicitly while other parts consist of calls to Web services that may be used to obtain more data or request some particular processing. We can see such documents as *intentional*, since parts in them are defined by programs that are used to obtain data when needed. We also view them as dynamic in the sense of dynamic Web pages that possibly return different documents when called at different time. An *AXML peer* consists of a repository containing (A)XML documents. It is a client in that activations of the calls inside the documents use Web services. It is a server in the sense that it provides queries to the repository as Web services.

A key aspect of the approach is that AXML peers exchange AXML documents, i.e., document with embedded service calls. Of particular importance is the process of *materializing* some or all the calls in a document and replacing them by their results.

We believe that the AXML approach is an appropriate infrastructure for the development of a P2P warehouse for a number of reasons:

1. AXML is based on a P2P architecture and uses the standards of the Web, and in particular, XML and Web services.
2. Each AXML peer provides an XML repository and an XML query processor.
3. Web services may be called from virtually anywhere, so warehouse functionalities may be published and used in a transparent manner.

Indeed, a first step in that direction is the SPIN project [7] that aims to constructing a warehouse from resources found on the Web using AXML. A lot of works is still needed in particular to incorporate aspects such as cooperative crawl or dynamic hash tables.

Acknowledgments I want to thank A. Milo for discussions on content warehouses, G. Cobena, A. Poggi B. Nguyen for discussions on SPIN, T. Milo for discussions on the ideas described in this paper, and P. Valduriez, T. Priol for general discussions on P2P data management. Last but not least, I would like to thank the entire AXML team for the fun we share in developing the language and the system.

References

1. K. Aberer, P-Grid: A Self-Organizing Access Structure for P2P Information Systems, CoopIS, 179-194, 2001.

2. The AXML project, INRIA,
http://www-rocq.inria.fr/verso/Gemo/Projects/axml.
3. S. Abiteboul, A. Bonifati, G. Cobéna, I. Manolescu, T. Milo, Active XML Documents with Distribution and Replication, ACM SIGMOD, 2003.
4. S. Abiteboul, P. Buneman, and D. Suciu, Data on the Web, Morgan Kaufmann Publishers, 2000.
5. S. Abiteboul, S. Cluet, G. Ferran and A. Milo, Xyleme Content Warehouse, Xyleme whitepaper, 2003.
6. S. Abiteboul, M. Preda, G. Cobena, Adaptive On-Line Page Importance Computation, WWW Conference, 2003.
7. S. Abiteboul, G. Cobena, B. Nguyen, A. Poggi, Construction and Maintenance of a Set of Pages of Interest (SPIN), Conference on Bases de Donnees Avancees, 2002.
8. H. Balakrishnan, M. F. Kaashoek, D. Karger, R. Morris, I. Stoica, Looking up data in P2P systems, CACM 46(2): 43-48, 2003.
9. T. Berners Lee, O. Lassila, and R. R. Swick, The semantic Web, Scientic American, vol. 5, 2001.
10. R. Braumandl, M. Keidl, A. Kemper, D. Kossmann, A. Kreutz, S. Seltzsam, K. Stocker, ObjectGlobe: Ubiquitous query processing on the Internet, The VLDB Journal, 10:48, 2001.
11. Gregory Cobéna, Serge Abiteboul, Amélie Marian, Detecting Changes in XML Documents, Proceedings of the Int'l ICDE Conference, 2002.
12. F. M. Cuenca-Acuna, C. Peery, R. P. Martin, T. D. Nguyen, PlanetP: Using Gossiping to Build Content Addressable Peer-to-Peer Information Sharing Communities, Department of Computer Science, Rutgers University, 2002.
13. T. Grabs, K. Böhm, H.-J. Schek, Scalable Distributed Query and Update Service Implementations for XML Document Elements, IEEE RIDE Int. Workshop on Document Management for Data Intensive Business and Scientific Applications, 2001.
14. F. Goasdoue and M.-C. Rousset, Querying Distributed Data through Distributed Ontologies: a Simple but Scalable Approach, 2003.
15. S. Gribble, A. Halevy, Z. Ives, M. Rodrig, D. Suciu, What can databases do for peer-to-peer? WebDB Workshop on Databases and the Web, 2001.
16. A. Y. Halevy, Z. G. Ives, D. Suciu, I. Tatarinov, Schema Mediation in Peer Data Management Systems, ICDE, 2003.
17. M. Harren, J. Hellerstein, R. Huebsch, B. Thau Loo, S. Shenker, I. Stoica, Complex Queries in DHT-based Peer-to-Peer Networks, Peer-to-Peer Systems Int. Workshop, 2002.
18. A. Kementsietsidis, M. Arenas, and R. Miller, Mapping Data in Peer-to-Peer Systems: Semantics and Algorithmic Issues, ACM SIGMOD, 2003.
19. M. Lenzerini, Data Integration, A Theoretical Perspective, ACM PODS 20'02, Madison, Winsconsin, USA, 2002.
20. T. Milo, S. Abiteboul, B. Amann, O. Benjelloun, F. Dang Ngoc, Exchanging Intensional XML Data, ACM SIGMOD, 2003.
21. B. Nguyen, S. Abiteboul, G. Cobena, M. Preda, Monitoring XML Data on the Web, ACM SIGMOD, 2001.
22. M.T. Özsszu, and P. Valduriez, Principles of Distributed Database Systems, Prentice-Hall, 1999.
23. The Piazza Project, U. Washington
http://data.cs.washington.edu/p2p/piazza/
24. J. Widom, Research Problems in Data Warehousing, In Proceedings of the 4th Int'l Conference on Information and Knowledge Management (CIKM), 1995.

25. F. M. Cuenca-Acuna, C. Peery, R. P. Martin, T. D. Nguyen, Using Gossiping to Build Content Addressable Peer-to-Peer Information Sharing Communities, Department of Computer Science, Rutgers University, 2002.
26. Xyleme Web site, http://www.xyleme.com

This article was processed using the LaTeX macro package with LLNCS style