

# A Calculus and Algebra for Distributed Data Management<sup>\*</sup>

Serge Abiteboul

INRIA-Futurs and U. of Paris, Orsay [fname.lname@inria.fr](mailto:fname.lname@inria.fr)

**Abstract.** The sharing of content by communities of users (e.g., scientists) in a P2P context remains cumbersome. We argue that main reasons for this is the lack of calculus and algebra for distributed data management. We present the ActiveXML language that extends the XML language with features to handle distribution. More precisely, ActiveXML documents are XML documents with a special syntax for specifying the embedding of Web service calls, e.g. XML queries such as XQueries. We also present ActiveXML algebra that extends ActiveXML notably with explicit control of data exchanges. ActiveXML algebra allows describing query plans, and exchanging them between peers.

**Keywords.** distributed data management, XML, Web, Web service, calculus, algebra, monitoring, stream.

The field of distributed data management [17] has centered for many years around the relational model. More recently, the Web has made the world wide and intranet publication of data much simpler, by relying on HTML, Web browsers, plain-text search engines and query forms. The situation has also dramatically improved with the introduction of XML [22] and Web services [25]. Together, these two standards provide an infrastructure for distributed computing at large, independent of any platform, system or programming language, i.e., the appropriate framework for distributed management of information. However, the sharing of content by communities of users (e.g., scientists) in a P2P context remains cumbersome. We argue that main reasons for this is the lack of calculus and algebra for distributed data management and propose such languages based on Web standards, namely XML and Web services.

In [8], we propose the *data ring* that can be seen as a network analogue of a database or a content warehouse. The vision is to build a P2P middleware system that can be used by a community of non-experts, such as scientists, to build content sharing communities in a declarative fashion. Essentially, a peer joins a data ring by specifying which data (or services in general) are to be shared, without having to specify a schema for the data, load it in a store, create any indices on it, or specify anything complex regarding its distribution.

---

<sup>\*</sup> This work has been partially supported by the ANR Project WebContent and the EC project Edos on the development and distribution of open source software.

The data ring enables users to perform declarative queries over the aggregated data, and becomes responsible for reorganizing the physical storage of data and for controlling its distribution. Thus a primary focus of the data ring is *simplicity of use*. To achieve these goals, we identified a number of challenges:

**Self-administration** Since the users of the data ring are non-expert, the deployment of the ring and its administration should be almost effort-less. This means that a number of tasks such as the selection of access structures (indices) or the gathering of the statistics to be used by optimizers have to be fully automatic.

**File management** Since a large part of the data is going to reside in file systems, we need very efficient processing and *optimization* of queries over files, including for instance the automatic selection of specific access structures over file collections.

**Query language** To facilitate the exploitation of the ring by non-experts, the interfaces have to be mostly graphical and require the minimum expertise. They therefore must be based on declarative languages (calculus) in the style of relational calculus, rather than on languages such as Java or Ajax that require programming skills.

**Query optimization** Query optimization has, by nature, to be distributed and peers should be able to exchange query plans. This motivates adopting an algebra for describing distributed query plans interleaving query optimization, query evaluation, and possibly, error recovery and transaction processing.

We present ActiveXML, a declarative framework that harnesses XML and Web services for the integration and management of distributed data. An ActiveXML document is an XML document where some of the data is given explicitly, while other portions are given only intensionally by means of embedded calls to Web services, typically XML queries. By calling the services, one can obtain up-to-date information. In particular, ActiveXML provides control over the activation of service calls both from the client side (pull) or from the server side (push).

It should be noted that the idea of mixing data and code is not new, e.g., stored procedures in relational systems [19], method calls in object-oriented databases [10], and queries in scripting languages such as PHP. The novelty is that since both XML and Web services are standards, ActiveXML documents can be universally understood, and therefore can be universally *exchanged*.

We also present the ActiveXML algebra that extends ActiveXML in two main directions: (i) with generic services that can be supported by several peers (e.g., query services), (ii) with explicit control of the evaluation of ActiveXML documents (eval operator) and of data exchange (send and receive operators). The algebra can be used to describe query (evaluation) plans. Using rewrite rules, query plans may be optimized in a standard way. More fundamentally, the query plans are distributed and can be exchanged between peers. Thus the tasks of query evaluation and optimization can be distributed among the peers of the network.

The ActiveXML project has been going on for several years. A system is now available as open source [9]. In [16], a technique to decide whether or not calls should be activated based on typing is introduced. The general problem has deep connections with tree automata [12] and alternating automata, i.e., automata alternating between universal and existential states [18]. Optimization issues in the context of ActiveXML are presented in [2]. In [5], a framework for managing distribution and replication in the context of ActiveXML is considered. Foundations of ActiveXML are studied in [3]. A preliminary version of the algebra appeared in [7].

## References

1. S. Abiteboul, Z. Abrams, S. Haar, and T. Milo. Diagnosis of asynchronous discrete event systems - Datalog to the rescue! In *ACM PODS*, 2005.
2. S. Abiteboul, O. Benjelloun, B. Cautis, I. Manolescu, T. Milo, N. Preda, Lazy Query Evaluation for Active XML, In Proc. of ACM SIGMOD 2004.
3. S. Abiteboul, O. Benjelloun, T. Milo, Positive Active XML, In Proc. of ACM PODS, 2004.
4. S. Abiteboul, P. Buneman, D. Suciu, Data on the Web, Morgan Kaufmann, 2000.
5. S. Abiteboul, A. Bonifati, G. Cobena, I. Manolescu, T. Milo, Active XML Documents with Distribution and Replication, In Proc. of ACM SIGMOD, 2003.
6. S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, Reading-Massachusetts, 1995.
7. Abiteboul, S., I. Manolescu, E. Taropa. A framework for distributed XML data management. In *Proc. EDBT*. 2006.
8. Serge Abiteboul, Neoklis Polyzotis, The Data Ring: Community Content Sharing In Proceedings of CIDR, 2007.
9. The ActiveXML project, INRIA, <http://activexml.net>.
10. The Object Database Standard: ODMG-93, editor R. G. G. Cattell, Morgan Kaufmann, San Mateo, California, 1994.
11. Sharma Chakravarthy, Jennifer Widom: Foreword: Special Issue on Active Database Systems. *J. Intell. Inf. Syst.* 7(2): 109-110. 1996.
12. H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, D. Lugiez, S. Tison, M. Tommasi, Tata, Tree Automata Techniques and Applications, [www.grappa.univ-lille3.fr/tata/](http://www.grappa.univ-lille3.fr/tata/)
13. The Edos Project, <http://www.edos-project.org/>
14. Laura M. Haas, Donald Kossmann, Edward L. Wimmers, and Jun Yang. Optimizing Queries Across Diverse Data Sources. In *vldb97*, pages 276–285, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.
15. M. Harren, J. Hellerstein, R. Huebsch, B. Thau Loo, S. Shenker, and I. Stoica. Complex queries in dht-based peer-to-peer networks. In *Peer-to-Peer Systems Int. Workshop*, 2002.
16. T. Milo, S. Abiteboul, B. Amann, O. Benjelloun, F. Dang Ngoc, Exchanging Intensional XML Data, In Proc. of ACM SIGMOD, 2003.
17. M.T. Ozsu, P. Valduriez, Principles of Distributed Database Systems, Prentice-Hall, 1999.
18. A. Muscholl, T. Schwentick, L. Segoufin, Active Context-Free Games, Symposium on Theoretical Aspects of Computer Science, 2004.

19. J.D. Ullman, Principles of Database and Knowledge Base Systems, Volume I, II, Computer Science Press, 1988.
20. The SOAP Specification, version 1.2, <http://www.w3.org/TR/soap12/>
21. Universal Description, Discovery and Integration of Web Services (UDDI), <http://www.uddi.org/>
22. The Extensible Markup Language (XML), <http://www.w3.org/XML/>
23. XML Typing Language (XML Schema), <http://www.w3.org/XML/Schema>
24. An XML Query Language, <http://www.w3.org/TR/xquery/>
25. The W3C Web Services Activity, <http://www.w3.org/2002/ws/>
26. The Web Services Description Language (WSDL), <http://www.w3.org/TR/wsdl/>