

Temporal Information Management using XML

Fusheng Wang, Xin Zhou, and Carlo Zaniolo

Department of Computer Science, University of California, Los Angeles,
Los Angeles, CA 90095, USA
{wangfsh,xinzhou,zaniolo}@cs.ucla.edu

1 Introduction

A closer integration of XML and database systems is actively pursued by researchers and vendors because of the many practical benefits it offers. Additional special benefits can be achieved on temporal information management— an important application area that represents an unsolved challenge for relational databases [10]. Indeed, XML data model and query languages support:

- Temporally grouped representations that have long been recognized as a natural data model for historical information [7], and
- Turing-complete query languages, such as XQuery [4], where all the constructs needed for temporal queries can be introduced as user-defined libraries, without requiring extensions to existing standards.

By contrast, the flat relational tables of traditional DBMSs are not well-suited for temporally grouped representations [8]; moreover, significant extensions are required to support temporal information in SQL and, in the past, they were poorly received by SQL standard committees.

We will show that (i) XML hierarchical structure can naturally represent the history of databases and XML documents via temporally-grouped data models, and (ii) powerful temporal queries can be expressed in XQuery without requiring any extension to current standards. This approach is quite general and, in addition to the evolution history of databases, it can be used to support the version history of XML documents for transaction-time, valid-time, and bitemporal chronicles [13]. We will demo the queries discussed in [13] and show that this approach leads to simple programming environments that are fully-integrated with current XML tools and commercial DBMSs.

2 The Systems ArchIS and ICAP

In our demo, we first show that transaction-time history of relational databases can be effectively published as XML views, where complex temporal queries on the evolution of database relations can be expressed in standard XQuery [11]. Therefore, we will demonstrate our ArchIS prototype that supports these queries efficiently on traditional database systems enhanced with SQL/XML. A temporal library of XQuery functions is used to facilitate the writing of the more complex queries and hide some implementation details (e.g., the internal

representation of ‘now’). We can thus support the complete gamut of historical queries, including snapshot and time-slicing queries, element-history queries, *since* and *until* queries. These temporal queries in XQuery are then mapped and executed as equivalent SQL/XML queries inside the RDBMS.

The next topic in the demo will be the application of our temporal representations and queries to XML documents of arbitrary nesting complexity. In the ICAP project [1], we store the version history of documents of public interest in ways that assure that powerful historical queries can be easily expressed and supported. Examples include successive versions of standards and normative documents, such as the UCLA course catalog [2], and the W3C Xlink specs [3], which are issued in XML form. Toward this objective,

- (i) we use structured diff algorithms [6, 5, 9] to compute the validity periods of the elements in the document,
- (ii) we use the output generated by the diff algorithm, to generate concisely the history of the documents with a temporally grouped data model. Then, on this representation,
- (iii) we use XQuery, enhanced with the library of temporal functions discussed above, to formulate temporal queries on the evolution of these documents and their contents.

The ICAP system also provides additional version-support services, including the ability of color-marking changes between versions, and annotating the changes with explanations on the changes and their reasons.

References

1. UCLA ICAP Project. <http://wis.cs.ucla.edu/projects/icap/>.
2. UCLA Catalog. <http://www.registrar.ucla.edu/catalog/>.
3. XML Linking Language (XLink). <http://www.w3.org/TR/xlink/>.
4. XQuery 1.0: An XML Query Language. <http://www.w3.org/TR/xquery/>.
5. Microsoft XML Diff. <http://apps.gotdotnet.com/xmltools/xmldiff/>.
6. S. Chawathe, A. Rajaraman, H. Garcia-Molina, J. Widom, “*Change Detection in Hierarchically Structured Information*”, in SIGMOD 1996.
7. J. Clifford. *Formal Semantics and Pragmatics for Natural Language Querying*. Cambridge University Press, 1990.
8. J. Clifford, A. Croker, F. Grandi, and A. Tuzhilin, “*On Temporal Grouping*”, in Proc. of the Intl. Workshop on Temporal Databases, 1995.
9. Gregory Cobena, Serge Abiteboul, Amelie Marian, “*Detecting Changes in XML Documents*”, in ICDE 2002.
10. G. Ozsoyoglu and R.T. Snodgrass, “*Temporal and real-time databases: A survey*”. in TKDE, 7(4):513–532, 1995.
11. F. Wang and C. Zaniolo, “*Publishing and Querying the Histories of Archived Relational Databases in XML*”, in WISE 2003.
12. Y. Wang, D. J. DeWitt, and J. Cai, “*X-Diff: A Fast Change Detection Algorithm for XML Documents*”, in ICDE 2003.
13. F. Wang and C. Zaniolo. “*XBiT: An XML-based Bitemporal Data Model*”, in ER 2004.