Time Versus Standards: A Tale of Temporal Databases

Carlo Zaniolo

Computer Science Department
University of California, Los Angeles, CA
zaniolo@cs.ucla.edu

Abstract. Because of the importance of time in Information Systems, there is a wide consensus on the need for having temporal data types and temporal queries supported as part of database standards. Actually, in the mid 90s, a forward-looking group of database researchers proposed an ambitious temporal extension of SQL-2 named TSQL2. Unfortunately, this proposal was ahead of its time and was not well-received by standard committees and DBMS vendors represented in those committees. Since then, standardization groups have not spent much effort on temporal issues, and, hence, it is natural to assume that no progress was made, and the gap between database standards and temporal applications remains as wide as it was in the mid-90s. This would represent very bad news for temporal application developers who suffer from inadequacies of database standards and do not see any relief in the near future. Fortunately, time conquers all and has also transformed standards, making them more supportive of time-oriented applications, in ways that are not fully recognized even by those who, motivated by different goals, pushed for the introduction of the new features.

The first significant rapprochement between standards and temporal applications occurred with the introduction of XML and XQuery, which, respectively, support a temporally grouped representation, and provide a powerful and extensible query language. These enable users to effectively model temporal information and pose complex historical queries without requiring changes in the standards. For all the merits of XML, its limited performance and scalability, combined with unwavering loyalty of users to relational data models and query languages, suggests that SQL will remain the crucial testbed for a successful rendezvous of standards and time. In this context, the SQL:2003 standards mark a significant step forward because of their OLAP Functions, which entail efficient expressions of temporal aggregates, and other important temporal functions, including coalescing. An even more exciting development is represented by the recently proposed new SQL standards for finding patterns in sequences of rows. By employing these Kleene-closure based constructs on suitable representations, users can then express very interesting temporal queries in SQL. Moreover, the power and flexibility of these extensions overcome the rigid division between event-based temporal data models and state-based models—a very desirable unification since, in the real world, a salary-increase event and the state of being at the new salary level are viewed as two facets of the same history.

I.-Y. Song et al. (Eds.): ER Workshops 2008, LNCS 5232, p. 67, 2008.