

The widespread use of XML for describing and exchanging data on the web, together with increasing quantities of XML data in the enterprise world, make it crucial to have an efficient query capability for XML data. In anticipation of an XML-rich world, the W3C XML Query Working Group has given us XQuery, a declarative query language designed specifically for XML. XQuery 1.0 became an official W3C standard (a Recommendation) in January 2007. Today, XQuery is gaining traction in industry; most major relational database systems, including products from IBM, Oracle, and Microsoft, now support an XML data type and include XQuery support for querying the contents of XML columns of tables. In middleware, leading enterprise service buses support both XQuery and XSLT for data transformation and routing, and several information integration products are based on XML and XQuery as well. XML, and to an extent XQuery, is also starting to penetrate key industry sectors (such as publishing, government, and pharmaceuticals) that have heavy technical document management requirements.

At the same time, XQuery itself is evolving. The W3C XML Query working group has several XQuery extension activities in progress. These include XQuery 1.1, extending the capabilities of XQuery with features such as grouping and aggregation; the XQuery Update Facility, adding XML update functionality to XQuery; the XQuery Full-Text Extension, adding content-based query capabilities to XQuery for text-heavy XML data; and, last but not least, the XQuery Scripting Facility, adding the ability to mix procedural-style control and side effects with the declarative query capabilities of XQuery in support of complex, XML-centric applications.

These developments make this a good time to take a look at the current state of the art in the XQuery processing world, both from academic and industrial perspectives. That is the purpose of this special issue of the *Data Engineering Bulletin*. This issue presents a snapshot of the current state of both the art and the practice of XQuery processing. Due to space limitations and busy potential contributors, the snapshot is of course incomplete, but we feel we have captured an interesting range of XQuery processors. When considering applications of XQuery, one finds a broad range of potential use cases that range from file processing and data transformation to message processing, to data integration, and to data and/or document storage, management, and querying. We have attempted to cover the full range in assembling this special issue.

The first two articles discuss systems (*Pathfinder* and *TIMBER*) developed in academia, using two very different approaches, one based on a relational implementation and one on native XML storage. The next three articles describe server-side XQuery processors. The articles on IBM's DB2 *pureXML* and Oracle's *XML DB* deal with XQuery in relational data servers, while the third examines XQuery in a native XML server (*MarkLogic Server*) aimed at content-oriented XML use cases. Next in this special issue are two articles about middleware XQuery processors, namely, the XQuery engine from BEA's *AquaLogic Data Services Platform* and the *DataDirect XQuery* engine, each of which use XQuery for information integration. The final article is about XQuery processing in *Saxon*, a leading open source XQuery engine.

We hope that this special issue will serve as a starting point for further academic and industrial contributions, as XQuery's increasing acceptance and ongoing evolution provide a fertile ground for interesting new research. We would like to thank the articles' authors, all experts in the field, for their timely efforts in assembling their excellent contributions for this special issue. Special thanks go to Marcos Vieira at UC Riverside for editorial assistance with the issue.