Foreword

There are at least two reasons why I am particularly glad to have the opportunity to write a foreword to this book. The first reason is a personal one: it has to do with my fond memory of the exchanges of ideas I had with the book's editor, Eric Pardede, in 2002, when I was visiting at LaTrobe University. Even then, it was easy to see that Eric was to become a rising star in the field of XML-related research.

The second reason has to do with the book's title, "Open and Novel Issues in XML Database Applications: Future Directions and Advanced Technologies", and contents. Indeed, reading this book, one can fully understand how much XML-related research has changed the entire database field in the last decade.

Ten years ago, XML data items were still called "tagged documents"; in the best case, they were seen as instances of nested semi-structured data (Abiteboul, S., 1997). Indeed, XML was born already equipped with a loose notion of schema, the Document Type Definition (DTD), but in the late 1990s many researchers considered DTDs as little more than patterns that XML documents should match. Some experts even forecasted that most XML data would be produced without a DTD, leaving it to extractors based on formal languages theory to infer formal languages (the DTDs) from examples (the XML documents).

It took a couple of years before a clear notion of XML data as typed, semi-structured information complying with a schema, began to emerge. The notion of an XML data model proved to be of great interest for database research, since it allowed it to finally conquer the "no man's land" between traditional database models and semi-structured information. Up until that moment, that "no man's land" had successfully resisted the combined onslaught of traditional approaches to databases and of semi-structured data processing techniques.

On the one hand, structured DBMSs (RDB or OODB) cannot easily handle flat streams or text blobs, and need additional components to do so (Abiteboul, S., Cluet, S., & Milo, T., 1998); on the other hand, traditional semi-structured data handling techniques cannot always guarantee time efficiency in answering queries and space efficiency in storing information.

The XML format, instead, could support efficient storage and fast querying of diversely structured data sets. Once this was realized, much effort went into developing XML query languages, their indexing support, and their execution algorithms. Indeed, some researchers (and a few software vendors) had, for a while, maintained the latter to be unnecessary, provided that XML storage was implemented on top of a relational databases. Early evidence, however, suggested that reconstructing XML data from relational ones could introduce a substantial overhead (e.g., from computing joins to obtain XML nested structure); also, some extra effort might be needed to translate XML queries and updates into SQL (Florescu, D., & Kossmann, D., 1999).

In the early 2000s, native XML repositories seemed the natural answer to these problems, as well as to some other issues related to the performance of concurrent access to XML data. Research into native XML repositories was aimed at identifying the right level of granularity per storage unit (an element, a sub-tree or an entire XML document) in order to support effective indexing and efficient queries. XML query languages, their syntax and semantics (Comai, S., Damiani, E., & Fraternali, P., 2001) were proposed, discussed and finally standardized.

While this work was being done, it became clear that most software applications were using XML as a format for data-in-transit rather than for data-at-rest, and the notion of XML queries and views (e.g., as ways to export XML data from relational storage) became again an important one (Fernandez, M. F., Tan, W. C., Suciu, D., 2000). Our own work (Damiani, E., Capitani di Vimercati, S. D., Paraboschi, S., & Samarati, P., 2000) focused on showing how, while these data were exported, one could apply policy-driven transformations, filtering the XML data stream according to the data owner's access control policy.

Today, more than a decade after XML was first announced, its importance as one of the core technologies of the global, service-oriented Internet cannot be underestimated. Things have progressed greatly thanks to the database research community and to the W3C standardization groups. We now have a clear notion of XML Schema and a well-specified set of query and update languages.

Still, XML is a very lively research field and a number of crucial research problems related to XML remain to be solved.

The contributions collected by Eric Pardede in this edited book successfully tackle some of these XML research issues. These chapters belong to the third and most recent generation of XML research, and are authored by leading international researchers in the XML field. They deal with new advanced topics such as XML schema evolution, personalized access to XML data, hybrid XML-relational databases, XML data integration and security issues, as well as XML query processing and optimization, benchmarking, XML warehousing, differential encryption, and XML programming.

I recommend this book, rigorous yet accessible, to non-specialists and to all those operating in the industry and in academia who are interested in XML research. The chapters in this book are also recommended reading to all young Ph.D. and post-doc researchers in computer science who look for open problems and new research challenges. Indeed, these chapters give a clear idea of why the second decade of XML research promises to be even more challenging and rewarding than the first.

Ernesto Damiani
DTI – University of Milan, Italy

REFERENCES

Abiteboul, S. (1997). Querying semi-structured data. In *ICDT*, (pp. 1-18).

Abiteboul, S., Cluet, S., & Milo, T. (1998). A logical view of structured files. In *VLDB Journal, 7*(2), 96-114.

Comai, S., Damiani, E., & Fraternali, P. (2001). Computing graphical queries over XML data. *ACM Trans. Inf. Syst. 19*(4), 371-430.

Damiani, E., Capitani di Vimercati, S. D., Paraboschi, S., & Samarati, P. (2000). Design and implementation of an access control processor for XML documents. In *Computer Networks 33*(1-6), 59-75.

Fernandez, M. F., Tan, W. C., Suciu, D. (2000). SilkRoute: trading between relations and XML. In *Computer Networks 33*(1-6), 723-745.

Florescu, D., & Kossmann, D. (1999). Storing and Querying XML Data using an RDMBS. In *IEEE Data Engineering Bulletin 22(3)*, 37-42.

Ernesto Damiani *is a professor at the Department of Information Technology, University of Milan, where he leads the Software Architectures Lab. Professor Damiani holds/has held visiting positions at several international institutions, including George Mason University (Fairfax, VA, US) and LaTrobe University (Melbourne, Australia). Professor Damiani is an adjunct professor at the Sydney University of Technology (Australia). He has written several books and filed international patents; also, he co-authored more than two hundred research papers on advanced secure service-oriented architectures, open source software and business process design, software reuse and Web data semantics. Professor Damiani is the vice chair of IFIP WG 2.12 on Web Data Semantics and the secretary of IFIP WG 2.13 on Open Source Software Development. He coordinates several research projects funded by the Italian Ministry of Research and by private companies including Siemens Mobile, Cisco Systems, ST Microelectronics, BT Exact, Engineering, Telecom Italy and others.*