

Foreword

In 1967, William W. Tunnicliffe articulated the idea of separating the formatting of a document from its content. Based on this generic coding, Charles Goldfarb, Edward Mosher, and Raymond Lorie developed the Generalized Markup Language (GML). Goldfarb used their surnames to make up the term GML. GML markups (tags) can be used to define the structure of documents (e.g., headers, chapters) without further specifying how particular elements of this structure are to be presented. In 1986, the Standard Generalized Markup Language became an ISO standard. However, its complexity has prevented its widespread application.

In the late 1990s, an SGML subset referred to as the Extensible Markup Language (XML) appeared and soon became a W3C standard in 1999. XML is simpler to parse and process than full SGML and has been widely adopted since then as a format to exchange data. Despite its name, XML is not a language with a given vocabulary but rather a set of syntax rules that can be applied to create languages. Such languages are called applications of XML. Since its standardization, a plethora of XML applications have been developed by industry and academia alike. Today, the Extensible Markup Language is ubiquitous in computing and an indispensable building block in countless software systems.

There are numerous explanations tendered for the success of XML. Certainly, XML's basic simplicity allowed for its rapid acceptance. The original XML specification can be printed on only 30 pages compared to the more than 150-pages SGML specification from which it was derived. Second, when SGML came out, in-memory implementations were prohibitively expensive while alternative stream processing approaches proved to be significantly less useful. Since then, the luxury of using XML has become affordable with the number of available DRAM bits per dollar doubling every two years from 10^5 in 1986 to 10^7 bits per dollar in 1999. Eventually, XML documents are both intelligible for humans and machine readable, which makes it relatively easy to do something with data you get in XML. In particular, one common XML parser can be used to extract the relevant information items of XML documents for all conceivable kinds of XML applications.

A variety of other fundamental applications and supporting technologies build on XML. XML Schema Definition (XSD) is a schema definition language to constrain the content model of an XML instance document to a specific hierarchical element structure and particular element data types. Several other W3C standards such as the Resource Description Framework (RDF)—a general method for the formal description of logical statements about resources—and the Web Ontology Language (OWL)—a knowledge representation language for authoring ontologies—are based on XML and XSD. Custom applications and application specific tools can in turn be built upon these standards.

This book presents in three sections a multifaceted insight into compelling XML applications and conclusive technologies. Section 1 introduces new XML applications and presents approaches to ease

the use of existing XML-based standards. Section 2 is concerned with analyzing and ensuring the quality of XML-based learning objects and repositories. Section 3 covers the spectrum from XML to databases and ontologies.

In Section 1, two chapters focus on improving the ease of use of XML by facilitating the generation-by-example of XSLT programs and XQuery expressions. In addition, the PExIL XML application provides for comprehensive descriptions of programming language exercises. Finally, an approach is presented to facilitate the privacy-aware sharing of personal data on the Web by means of an instant messaging protocol.

In Section 2, multi-criteria analysis and optimization methods for the evaluation and the improvement of learning repositories are explored. In addition, a framework is presented for quality assessments of the specifications of learning objects and repositories. Moreover, techniques for the detection of mobile devices are compared followed by the introduction of an architectural model for the delivery of XML-based mobile-learning content.

In Section 3, two chapters focus on database preservation using XML-based technologies. First, the SIARD standard for Software Independent Archival of Relational Databases is extended to include metadata specific to data warehouses. Second, the Web Ontology Language is used to describe conceptual models of relational databases in order to facilitate archival storage. Furthermore, a complete workflow for digital curation based on XML documents is presented. A combination of XML's contextual modeling and ontologies is described in the context of information access for the website of the Portuguese Emigration Museum.

The chapters in this book are extended versions of selected papers of the Ninth XATA Workshop (2011), which was held at the Escola Superior de Estudos Industriais e de Gestão at the Instituto Politécnico do Porto in Portugal. I had the privilege to serve as a member of the editorial advisory board and can warmly recommend this volume as an inspiring compilation of recent approaches in the area of XML and related technologies. At the same time, I suggest that interested readers may consider attending or submitting to the recently established Symposium on Languages, Applications, and Technologies (SLATE), which continues with a dedicated track to cover the topics of the former XATA workshop series.

Computing was not ready when SGML came out. Today we are ready for XML. Enjoy!

Alexander Paar
University of Pretoria, South Africa

Alexander Paar is a Software Developer, Project Manager, and Computer Scientist. He studied Computer Science at the Clausthal University of Technology, the University of California, Irvine, and the University of Karlsruhe (TH). He earned a Computer Science Doctorate from the University of Karlsruhe (TH) in 2009, then was a post-doctoral research fellow at the University of Pretoria. He contributed to international research projects from conception to completion and has a proven record as a speaker at international computer science workshops and conferences. He served as a program committee member and reviewed submissions for various computer science workshops, conferences, and journals. Alexander authored the Zhi# programming language—the first object-oriented programming language with inherent support for ontologies. He was in second place in the 2001 IEEE CSIDC and won silver and bronze medals for coaching teams for the ACM ICPC. He is a member of ACM, IEEE CS, and Gesellschaft für Informatik.