



Preface

For nearly three thousand years information has been organized in the purposeful and self-contained package that we call a document. The technology for encoding and exchanging documents has profoundly changed, but the concept of a document has remained surprisingly stable. Documents formalize the interactions between enterprises and their customers or clients, and it natural and intuitive to view documents as the input requirements and as the output results from many kinds of processes. These document exchanges follow common patterns. Models of business organization like supply chains, business-to-business marketplaces, and auctions can be composed from simpler two-party patterns of document exchanges.

The Internet and its supporting technologies like XML and web services give us great efficiencies and flexibility in how we create, manage and share information to conduct business and collaborate with others. But taking full advantage of these new technologies requires that we continue to think of documents in an abstract and technology-neutral way. When a seller asks, “What will you order from my catalog?” or the buyer asks, “Will you take this check as payment?” they are focusing on what they want to accomplish through their interactions, not on how to do it. Similarly, enterprise applications that automate business processes involving document exchanges should expose their interfaces as abstract document models so they can operate in heterogeneous technology environments.

Document Engineering helps us specify, design, and implement these documents and the processes that create and consume them. It synthesizes complementary ideas from information and systems analysis, electronic publishing, business process analysis, and business informatics to ensure that the documents and processes make sense to the people and applications that need them. A document-centric philosophy unifies these different analysis and modeling perspectives. Using patterns for document exchanges and document components ensures we can build applications and services that are robust but adaptable when technology or business conditions change (as they inevitably will).

About the Authors

Bob Glushko spent many years in industrial research and development, technology transfer, and consulting with a focus that evolved from human factors in computing systems, to electronic publishing, and then to Internet commerce. He founded or co-founded three companies, the last of which was Veo Systems in 1997, which pioneered the use of XML for electronic commerce before its 1999 acquisition by

Commerce One. From 1999-2002 he headed Commerce One's XML architecture and technical standards activities and was named an "Engineering Fellow" in 2000. In 2002 he became an Adjunct Professor in the School of Information Management and Systems at the University of California, Berkeley where he is the Director of the Center for Document Engineering.

Tim McGrath has a background in information systems design, specifically in the area of trade and transport systems. In 1990 he co-founded Transport EDI Services (TEDIS), which grew to be a leader in innovative Internet services for EDI in Australia. Apart from spending the past three years writing this book he has been the chair of the Universal Business Language Library Content subcommittee.

Bob and Tim met in 2000 as members of the Quality Review Committee in the ebXML standards initiative. This committee exercised broad technical oversight over the entire suite of ebXML standards, including information and process models, methodologies, and technical architecture for Internet business applications.

Bob's work in Silicon Valley in the "new economy" of moving bits around the Internet was perfectly complemented by Tim's expertise in the "old economy" of moving real stuff around in the physical world. Bob's efforts with SGML and XML for document analysis were matched by Tim's with EDI and data modeling. It seemed natural to work together to create a coherent and comprehensive approach for Document Engineering that builds on their unique combination of perspectives and expertise.

Acknowledgments

Many people have contributed to this book. Lecture notes for the Document Engineering course at UC Berkeley became the outline for the first draft, and students in that and other courses read versions of many chapters. This final version of the book barely resembles those lecture notes and early drafts, which means that we received much useful feedback, but it also means that early generations of Berkeley students suffered at our hands and for that we apologize. In particular, we thank Patrick Garvey, Calvin Smith, Bill French, and Carolyn Cracraft for serving as the teaching assistants in Document Engineering courses. Students Kate Ahern, Alison Billings, Aaron Brick, Peter Charles, Bob Daly, Lisa de Larios-Heiman, Marc Gratacos, Denise Green, Kristine Gual, Ryan Huebsch, Sonia Klemperer-Johnson,

John Leon, Justin Makeig, Vam Makam, and Amy Todenhagen also contributed to improving the book.

We are grateful to Allison Bloodworth, Adam Blum, Ron Bourret, David Burdett, Hank Chesbrough, Larry Downes, Robert M. Glushko, Brian Hayes, Mary Loomis, Scott McMullan, Alex Milowski, Hari Reddy, Pamela Samuelson, Anno Saxenian, and Hal Varian for their careful reviews of draft chapters. Our book is much better because of their insights and frankness.

We thank Allison Bloodworth (again), Nadine Fiebrich, Myra Liu, and Zhanna Shamis for allowing us to use their Berkeley Event Calendar Network project as an extended case study.

The concepts and methods in this book partly evolved from our work in the ebXML and Universal Business Language initiatives. There are far too many individuals who have contributed to our ideas through these projects to name them all; we hope they will feel acknowledged when they see their collective insights in the text. But we would like to specifically recognize Joe Baran, Jon Bosak, Toufic Boubez, Bill Burcham, Stuart Campbell, Dave Carlson, Chin Chee-Kai, Klaus-Dieter Naujok, Stephen Green, Arofan Gregory, Eduardo Gutentag, Eve Maler, Duane Nickull, Sue Probert, Dick Raman, Mike Rawlins, Karsten Riemer, Marion Royal, Gunther Stuhec, David Webber, and last but by no means least, the late Mike Adcock.

Our work intersects with research at the University of Hong Kong and we would like to thank David Cheung and Thomas Lee from the Center for E-Commerce Infrastructure Development (CECID) for their valuable feedback and support.

We also thank Jon Conhaim, Paul Gray, Patrick McGrath, Helen Norris, JR Schulden and Shel Waggener of the University of California, Berkeley for their organizational and financial contributions to our work and especially for allowing us to use the e-Berkeley program and campus IT projects as a testbed for many of our ideas.

IBMers Sharon Adler, David Cohn, Rob Guttman, Paul Maglio, Bob Schloss, and especially Jim Spohrer inspired us by their own work in web services, business informatics, and services science to try to pull it all together here.

We couldn't have written this book without the indirect help of many colleagues with or from whom we learned much of what it is in it. These include Terry Allen, Liora Alschuler, Mike Bianchi, Peter Brown, Brian Caporlette, Ian Crawford, Matthew Fuchs, Clive Gregory, Sue Helper, Kevin Hughes, Russ Hunt, Ken Kershner, Eliot Kimber, Mary Laplante, Michael Leventhal, Tom Malone, John Mashey, John May, Bart Meltzer, Murray Maloney, Jeff Suttor, Bill Rouse, Marty Tenenbaum, Marcy Thompson, Ben Wolin and Vincent Vuong.

Loralee Windsor copyedited the book and Andrea Nelson designed its look and feel. Carolyn Cracraft created the index and glossary of terms. Naturally, any errors in content, structure or presentation are ours.

Doug Sery and MIT Press have been remarkably patient.

And finally, we want to thank Pamela Samuelson (again) and Isabelle, Hannah and Duncan McGrath for their self-sacrifice and inestimable encouragement to us while we wrote this book. Perhaps they knew better than we did what it would take, but never let on.

How This Book is Organized

This book is organized in four parts.

I "INTRODUCTION" is just that.

II "FOUNDATIONS" consists of five chapters that discuss XML, modeling, business patterns, and XML vocabularies to establish an intellectual baseline for the rest of the book. Some of this material will be familiar to practitioners but has proven essential for students, so we've separated it so that each can attend to it as needed.

III "THE DOCUMENT ENGINEERING APPROACH" begins with Chapter 7, which summarizes the end-to-end phases of Document Engineering. Each phase is treated in depth in a separate chapter. Chapter 16, "Management and Strategy," discusses considerations that span all of Document Engineering but which would be more difficult to explain if this chapter appeared earlier.

IV "THE END OF THE BEGINNING" contains a short epilogue, notes, glossary, and index.

We wrote this book for consultants, practitioners and advanced university students in information systems, industrial engineering, business informatics or related professional disciplines. The book strives to present enough theory and concepts to frame issues but is aggressively practical where the material allows us to be. The balance between theory and reality also shows in our notes, which are more extensive than in most business or trade books but which make no attempt to be as rigorous as those in academic literature. We also mix in archival and academic sources with web citations when the latter are likely to be more current or accessible. These notes appear in a separate section near the end of the book rather than at the bottom of each page or the end of each chapter so that readers are not confronted by them if they choose not to read them.

Applications, technologies, and issues for Document Engineering frequently appear in news stories and technical journals – but are not yet categorized that way. The companion web site for this book that collects and organizes them, and provides teaching materials and other useful resources is www.docengineering.com.

Bob Glushko (San Francisco, California)

Tim McGrath (Fremantle, Western Australia)