

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,