Foreword

Computers are changing the way we think. Of course, nearly all desk-workers have access to computers and use them to email their colleagues, search the Web for information and prepare documents. But I'm not referring to that. I mean that people have begun to think about what they do in computational terms and to exploit the power of computers to do things that would previously have been unimaginable.

This observation is especially true of mathematicians. Arithmetic computation is one of the roots of mathematics. Since Euclid's algorithm for finding greatest common divisors, many seminal mathematical contributions have consisted of new procedures. But powerful computer graphics have now enabled mathematicians to envisage the behaviour of these procedures and, thereby, gain new insights, make new conjectures and explore new avenues of research. Think of the explosive interest in fractals, for instance. This has been driven primarily by our new-found ability rapidly to visualise fractal shapes, such as the Mandelbrot set. Taking advantage of these new opportunities has required the learning of new skills, such as using computer algebra and graphics packages.

The argument is even stronger. It is not just that computational skills are a useful adjunct to a mathematician's arsenal, but that they are becoming essential. Mathematical knowledge is growing exponentially: following its own version of Moore's Law. Without computer-based information retrieval techniques it will be impossible to locate relevant theories and theorems, leading to a fragmentation and slowing down of the field as each research area rediscovers knowledge that is already well-known in other areas. Moreover, without the use of computers, there are potentially interesting theorems that will remain unproved. It is an immediate corollary of Gödel's Incompleteness Theorem that, however huge a proof you think of, there is a short theorem whose smallest proof *is* that huge. Without a computer to automate the discovery of the bulk of these huge proofs, we have no hope of proving these simple-stated theorems. We have already seen early examples of this phenomenon in the Four-Colour Theorem and Kepler's Conjecture on sphere packing. Perhaps computers can also help us to navigate, abstract and, hence, understand these huge proofs.

Realising this dream of computer access to a world repository of mathematical knowledge, visualising and understanding this knowledge, and reusing and combining it to discover new knowledge, presents a major challenge to mathematicians and informaticians. The first part of this challenge arises because mathematical knowledge will be distributed across multiple sources and represented in diverse ways. We need a lingua franca that will enable this babel of mathematical languages to communicate with each other. This is why this book — proposing just such a lingua franca — is so important. It lays the foundations for realising the rest of the dream.

OMDoc is an open markup language for mathematical documents. The 'markup' aspect of OMDoc means that we can take existing knowledge and annotate it with the information required to retrieve and combine it automatically. The 'open' aspect of OMDoc means that it is extensible, so future-proofed against new developments in mathematics, which is essential in such a rapidly growing and complex field of knowledge. These are both essential features. Mathematical knowledge is growing too fast and is too distributed for any centrally controlled solution to its management. Control must be distributed to the mathematical communities that produce it. We must provide lightweight mechanisms under local control that will enable those communities to put the produce of their labours into the commonwealth with minimal effort. Standards are required to enable interaction between these diverse knowledge sources, but they must be flexible and simple to use. These requirements have informed OMDoc's development. This book will explain to the international mathematics community what they need to do to contribute to and to exploit this growing body of distributed mathematical knowledge. It will become essential reading for all working mathematicians and mathematics students aspiring to take part in this new world of shared mathematical knowledge.

OMDoc is one of the first fruits of the Mathematical Knowledge Management (MKM) Network (`http://www.mkm-ig.org/`). This network combines researchers in mathematics, informatics and library science. It is attempting to realise the dream of creating a universal digital mathematics library of all mathematical knowledge accessible to all via the World-Wide-Web. Of course, this is one of those dreams that is never fully realised, but remains as a source of inspiration. Nevertheless, even its partial realisation would transform the way that mathematics is practised and learned. It would be a dynamic library, providing not just text, but allowing users to run computer software that would provide visualisations, calculate solutions, reveal counter-examples and prove theorems. It would not just be a passive source of knowledge but a partner in mathematical discovery. One major application of this library will be to teaching. Many of the participants in the MKM Network are building teaching aids that exploit the initial versions of the library.

There will be a seamless transition between teaching aids and research assistants — as the library adjusts its contribution to match the mathematical user's current needs. The library will be freely available to all: all nations, all age groups and all ability levels.

I'm delighted to write this foreword to one of the first steps in realising this vision.

May 2006 Alan Bundy

Preface

Mathematics is one of the oldest areas of human knowledge[1]. It forms the basis of most modern sciences, technology and engineering disciplines. Mathematics provides them with modeling tools such as statistical analysis or differential equations. Inventions like public-key cryptography show that no part of mathematics is fundamentally inapplicable. Last, but not least, we teach mathematics to our students to develop abstract thinking and hone their reasoning skills.

However, mathematical knowledge is far too vast to be understood by one person, moreover, it has been estimated that the total amount of published mathematics doubles every ten to fifteen years [Odl95]. Thus the question of supporting the management and dissemination of mathematical knowledge is becoming ever more pressing but remains difficult. Even though mathematical knowledge can vary greatly in its presentation, level of formality and rigor, there is a level of deep semantic structure that is common to all forms of mathematics and that must be represented to capture the essence of the knowledge.

At the same time it is plausible to expect that the way we do (i.e., conceive, develop, communicate about, and publish) mathematics will change considerably in the years to come. The Internet plays an ever-increasing role in our everyday life, and most of the mathematical activities will be supported by mathematical software systems connected by a commonly accepted distribution architecture, which makes the combined systems appear to the user as one homogeneous application. They will communicate with human users and amongst themselves by exchanging structured mathematical documents, whose document format makes the context of the communication and the meaning of the mathematical objects unambiguous.

Thus the inter-operation of mathematical services can be seen as a knowledge management task between software systems. On the other hand, math-

[1] We find mathematical knowledge written down on Sumerian clay tablets, and even Euclid's *Elements*, an early rigorous development of a larger body of mathematics, is over 2000 years old.

ematical knowledge management will almost certainly be web-based, distributed, modular, and integrated into the emerging math services architecture. So the two fields constrain and cross-fertilize each other at the same time. A shared fundamental task that has to be solved for the vision of a "web of mathematical knowledge" (MathWeb) to become reality is to define an open markup language for the mathematical objects and knowledge exchanged between mathematical services. The OMDoc format (Open Mathematical Documents) presented here is an answer to this challenge, it attempts to provide an infrastructure for the communication and storage of mathematical knowledge.

Mathematics – with its long tradition in the pursuit of conceptual clarity and representational rigor – is an interesting test case for general knowledge management, since it abstracts from vagueness of other knowledge without limiting its inherent complexity. The concentration on mathematics in OMDoc and this book does not preclude applications in other areas. On the contrary, all the material directly extends to the STEM (science, technology, education, and mathematics) fields, once a certain level of conceptualization has been reached.

This book tries to be a one-stop information source about the OMDoc format, its applications, and best practices. It is intended for authors of mathematical documents and for application developers. The book is divided into four parts: an introduction to markup for mathematics (Part I), an OMDoc primer with paradigmatic examples for many kinds of mathematical documents (Part II), the rigorous specification of the OMDoc document format (Part III), and an XML document type definition and schema (Part IV).

The book can be read in multiple ways:

– for users that only need a casual exposure to the format, or authors that have a specific text category in mind, it may be best to look at the examples in the OMDoc primer (Part II of this book),
– for an in-depth account of the format and all the possibilities of modeling mathematical documents, the rigorous specification in Part III is indispensable. This is particularly true for application developers, who will also want to study the external resources, existing OMDoc applications and projects, in Part IV.
– Application developers will also need to familiarize themselves with the OMDoc Schema in the Appendix.

Acknowledgments

Of course the OMDoc format has not been developed by one person alone. The original proposal was taken up by several research groups, most notably the ΩMEGA group at Saarland University, the Maya and ActiveMath projects at the German Research Center of Artificial Intelligence (DFKI), the MoWGLI EU Project, the RIACA group at the Technical University of Eindhoven, and the CourseCapsules project at Carnegie Mellon University. They discussed the initial proposals, represented their materials in OMDoc and in the process refined the format with numerous suggestions and discussions.

The author specifically would like to thank Serge Autexier, Bernd Krieg-Brückner, Olga Caprotti, David Carlisle, Claudio Sacerdoti Coen, Arjeh Cohen, Armin Fiedler, Andreas Franke, George Goguadze, Alberto González Palomo, Dieter Hutter, Andrea Kohlhase, Christoph Lange, Paul Libbrecht, Erica Melis, Till Mossakowski, Normen Müller, Immanuel Normann, Martijn Oostdijk, Martin Pollet, Julian Richardson, Manfred Riem, and Michel Vollebregt for their input, discussions, and feedback from implementations and applications.

Special thanks are due to Alan Bundy and Jörg Siekmann. The first triggered the work on OMDoc, has lent valuable insight over the years, and has graciously consented to write the foreword to this book. Jörg continually supported the OMDoc idea with his abundant and unwavering enthusiasm. In fact the very aim of the OMDoc format: openness, cooperation, and philosophic adequateness came from the spirit in his ΩMEGA group, which the author has had the privilege to belong to for more than 10 years.

The work presented in this book was supported by the "Deutsche Forschungsgemeinschaft" in the special research action "Resource-adaptive cognitive processes" (SFB 378), and a three-year Heisenberg Stipend to the author. Carnegie Mellon University, SRI International, and the International University Bremen have supported the author while working on revisions for versions 1.1 and 1.2.