Foreword

As a researcher in the field of software engineering, I am actively contributing to the scientific research in software evolution and model-driven software engineering. Being the chair of the Software Evolution Working Group of the European Research Consortium for Informatics and Mathematics (ERCIM), I have the honor to write the foreword for this book. It was a real pleasure for me to take up this task as this book aims to reconcile two extremely relevant topics in today’s software engineering research and practice: software evolution on the one hand, and model-driven software engineering on the other hand.

Software maintenance and software evolution have been considered as some of the most important challenges in software engineering since its origins in the late sixties. In fact, software maintenance was already coined as one of the crucial software engineering activities in the first NATO conference on software engineering in 1968. Since the early days, research progress in software evolution never ceased. Techniques, formalisms, tools, methods, and processes for software evolution slowly made it into mainstream programming practice, of which they are now an integral part. The general aim is to support software engineers in maintaining and evolving software-intensive systems in an effective manner without sacrificing quality. A fairly recent and comprehensive overview of the state-of-the-art in software evolution research can be found in the 2008 Springer book “Software Evolution.”

Model-driven software engineering has also been around for quite a while. Even in the aforementioned NATO conference, the importance of design and supporting techniques was clearly stated. Since then, an enormous amount of modeling languages and methodologies have been developed and used in a wide variety of domains (data models, software models, hardware models, domain-specific models, etc.). Somewhere in the nineties, model-driven software engineering gained a renewed widespread interest, due to the effort of the Object Management Group to standardize the many existing software modeling languages into a new unified modeling language, known as the UML. The popularity increased even more by the introduction of OMG’s Model-Driven Architecture method for developing software through the use of software models, and by annual scientific events such as the ACM/IEEE sponsored International Conference on Model Driven Engineering Languages and Systems (MoDELS) and Springer’s SCI-indexed scientific journal on Software and Systems Modeling (SoSyM).

From a practical point of view, software evolution support for model-driven development environments and methodologies is still lagging behind. In 2005, together with several co-authors, we identified the 18 most pressing “Challenges in Software Evolution”. These were presented in an article carrying the same name at IWPSE, the 8th International Workshop on Principles of Software Evolution (DOI: 10.1109/IWPSE.2005.7). Interestingly, “supporting model evolution” was identified as one of the key challenges. More precisely, the challenge was formulated as follows: “Software evolution techniques should be raised to a higher level of abstraction, in order to accommodate not only evolution of programs, but also evolution of higher-level artifacts such as analysis and design models, software architectures, requirements specifications, and so on.”
I am therefore very pleased that the aim of this book is to present the emerging research techniques for software model evolution and maintenance. Jörg Rech and Christian Bunse, editors of this book, have done an excellent job in collecting high-quality contributing chapters focusing on a wide range of different model evolution aspects:

• Model versioning, which aims to provide advanced version control support for software models
• Model-based testing, which aims to integrate automated tests in the model-driven software development process
• Model transformation to explore the use of transformation languages to evolve and manipulate software models
• Reusable models that help to develop new model-driven solutions through the reuse of existing modeling artifacts
• Model traceability and change impact analysis that allow to facilitate, and assess the effect of, propagating changes in models
• Model reverse engineering, which is used to reconstruct models from the program code, and these models can be used subsequently to re-engineer or modernize the system
• Model quality, which needs to be assured, preserved, and improved during model evolution
• As well as a wide variety of other emerging techniques that can aid in the model-driven development process, such as ontologies, social networks, and many more.

This book will provide a valuable source of information and inspiration for researchers and practitioners alike. Readers of this book will learn a lot about the interplay between the fields of software evolution and software modeling, and will get a good general picture of the current state-of-the-art.

I enjoyed reading this book, and I hope you will do so too. Happy reading!

Tom Mens
University of Mons, Belgium

Tom Mens obtained his PhD in Science in 1999 at the Vrije Universiteit Brussel (Belgium). After that, he was a postdoctoral fellow of the Fund for Scientific Research – Flanders (FWO) for three years. In October 2003 he became a Lecturer at the Faculty of Sciences of the University of Mons (Belgium), where he founded and directs a research lab on software engineering. Since November 2008 he is full Professor. His main research interest lies in the underlying foundations of, and tool support for, modeling, developing, and evolving software. He published numerous peer-reviewed articles on this research topic in international journals and conferences. He has been co-organizer, program committee member, and reviewer of numerous international symposia and workshops on model-driven software engineering and software evolution. He has been involved in several interuniversity research projects and networks, and is founder and chair of the ERCIM Working Group on Software Evolution. In 2008 he co-edited the Springer book “Software Evolution” with S. Demeyer.