Software Model Engineering has a long tradition of practice and innovation, probably as old as Program Engineering itself. The history of program engineering (in short programming) is more documented by excellent books and papers for several reasons. Since the first computers, one knows what a programming language is, and starting from early assembly languages, it is relatively easy to draw multiple maps of the evolution of these thousands of programming languages, insisting on various properties and various traceability and influence links. Unfortunately, the situation is not so clear when we look at the history of modeling languages. There is no unique banner under which we could classify the huge number of modeling formalisms that accompanied executable programming languages since the beginning. One easily remembers flowcharts used to prepare or to understand programs in the ages when these programs were still full of goto instructions. One could also talk about non-executable specifications, problem statement languages, structured analysis and design, formal and semi-formal methods, CASE formalisms, formal specifications, requirement languages, QOS formalisms, and many more. Today, we have a clearer view of all these modeling languages, mainly because we came to realize that they usually correspond to multiple Domain Specific Languages (DSLs). We can much easier understand the lineages linking PSL/PSA, JSD, SADT, HOOD, OMT, and many others. Understanding the history of modeling languages is of paramount importance to the present organization of research in the field. It allows us to understand when we are progressing, how we build on previous results, and where we could/should put additional efforts. This history is made of different phases, the last ones being the Object-Oriented Analysis and Design multiple proposals, the UML unification and standardization, the MDA initiative, and the MDE generalization. Each period has emphasized some specific features. The net result of this long evolution is probably that we have not yet reached the ultimate goal of having models everywhere in the software development cycle, but the demonstration of the tremendous potential of software modeling has now been firmly established. We are currently midstream, and it is particularly exciting to observe the constant flow of contributions that are offered to evolve and consolidate this important research field. This book *Progressions and Innovations in Model-Driven Software Engineering* is a typical illustration of this phenomenon. It not only shows the important number of projects going on in MDSE, but it also illustrates their high diversity. They apply to most moments of the software lifecycle from requirements, architecture, and code generation through validation and runtime until maintenance and evolution. They cover many application fields from electronic commerce and e-government to agricultural engineering. They address several techniques like standardization, performance evaluation, DSL development, executable requirements, metamodeling, and many more. They offer a typical snapshot of current MDSE initiatives and are a perfect illustration of the dynamism and diversity of research activities in this field. Keeping in mind the continuous and rapid evolution of software modeling theory and practices, it is a very pleasant and enriching reading experience that we are being offered by all chapters of this book.
Jean Bézivin is industry consultant in software modeling and emeritus professor of Computer Science at the University of Nantes. He previously created AtlanMod, a research team common to INRIA and Ecole des Mines de Nantes. He got his Master degree from the University of Grenoble and Ph.D. from the University of Rennes. He has been research assistant in 1972 at the Queen’s University of Belfast and in 1973 at the Concordia University of Montreal. Since 1980 he has been very active in Europe in the object-oriented and in the model-driven communities, helping to initiate several research conferences like ECOOP, TOOLS, UML/MODELS, ICMT, etc. He founded in 1979, at the University of Nantes, one of the first Master programs in Software Engineering entirely devoted to Object Technology (Databases, Concurrency, Languages and Programming, Analysis and Design, etc.). His research interests include model-driven engineering and more especially the techniques of model transformation applied to data engineering and to software forward and reverse engineering. He has published many papers and organized tutorials and workshops in the domains of concurrency, simulation, object-oriented programming, and model-driven engineering. On the subjects of model-driven engineering and MDA™, he has been leading the OFTA industrial group in France, co-animating a CNRS specific action, and a Dagstuhl seminar. He is a member of the ECOOP, TOOLS, MODELS, and ICMT steering committees. He was co-chair of the ECOOP 2006 conference organized in Nantes and program chair of TOOLS Europe 2007 held in Zurich in June 2007. He is currently interested in the broadening relations between software engineering and engineering software and in the multiple applications of software modeling to various domain engineering fields.

Jean Bézivin
University of Nantes, France