Modeling with UML

Bernhard Rumpe

Modeling with UML

Language, Concepts, Methods

Springer

Bernhard Rumpe
Software Engineering
RWTH Aachen University
Aachen
Germany

ISBN 978-3-319-33932-0 ISBN 978-3-319-33933-7 (eBook)
DOI 10.1007/978-3-319-33933-7

Library of Congress Control Number: 2016940125

Translation from the German language edition: Modellierung mit UML – Sprache, Konzepte und Methodik by B. Rumpe, © Springer-Verlag Berlin Heidelberg 2004, 2011. All Rights Reserved.

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG Switzerland

Foreword[1]

Designing large software systems is one of the big technical challenges of our time. The scope and complexity of software have now reached dimensions that push all established approaches and methods for its development to its limits.

In this situation, software developers have increasingly discovered the established concept of model creation in the engineering sciences. In the past, a large number of different approaches have worked out under the concept model-based software development, which aims at extensive model creation to support development of software systems. Model creation enables specific representations of important properties and aspects of a software system to be analyzed or designed. One objective is an appropriate abstraction leading to decreased complexity and improved controllability of software systems. Despite all the progress made in this field and its clear practical maturity, there are still many questions that need to be answered by research.

The additional development effort required is certainly a critical factor in model creation. The question here is how much effort should be invested in model creation and how model-based procedures, which are often heavyweight, can be made flexible enough to better consider the profiles of the development projects.

Besides model orientation, use of so-called agile methods has become another trend in software engineering in recent years, especially around the concept of "Extreme Programming". This term encompasses lightweight process models for software development that secure a reduction of software bureaucracy and support a much greater flexibility in software development. For projects with a certain profile, agile methods can facilitate a considerably more effective process. However, preconditions for this are sufficiently competent developers as well as a clearly limited project size. Thus, such agile methods can only be used successfully in small projects with a handful of developers over a manageable period of time so that feedback can actually work to achieve faster communication within the project.

[1]Translated from the Foreword of the German Edition.

At first sight, it seems that model-based approaches, with their strong systematics and their modeling techniques explicitly detached from the actual coding, are not compatible with agile methods, which are usually code-centered. This book impressively shows that it is still possible to combine model-based approaches with agile methods by using well-known modeling languages such as UML. However, one must then carefully consider which UML constructs can be used as modeling, testing, and implementation description tools and what the methodical procedure should look like.

This book provides an answer to this question, aiming to use relevant practical approaches such as the agile approach and the widespread language UML without leaving out a proper scientific foundation and well-documented process. In particular, it is clearly shown which UML constructs are suitable for, e.g., rigorously developing test cases or launching an evolution by applying perfect transformation rules.

The book demonstrates how the quite different paradigms of agile methods and model orientation correspond to and supplement each other. The result is an approach that equally satisfies the requirements for a practically relevant, well-usable procedure as well as the demand of a precise scientific foundation.

This text reads very well without giving up the claim of providing a solid content and technical representation. Bernhard Rumpe has successfully tested the process suggested in this book in a number of smaller projects.

Thus, this work represents a valuable contribution, providing useful guidance for practitioners and additional information on how to combine current trends in software engineering—such as agile procedures and model-based development—successfully and with reasonable additions. Students will receive a comprehensive introduction to the topic, and the book serves as a sound foundation.

This, as well as the consecutive book “Agile Modeling with UML” are equally well suited for practitioners interested in such an approach for their development projects as well as for lectures dealing with practical questions while not neglecting a fundamental scientific foundation.

Garching, Germany Manfred Broy
February 2004

Preface to the Second Edition[2]

Ten years ago, it could be foreseen that agile methods would prevail, at least for a substantial subdomain of software development, even though they were smiled at by many developers at that time. Today, agile methods have become an established part of the software engineering portfolio. In many places, they have been extended and adjusted to specific domains.

At the same time, the Unified Modeling Language started its triumph and has since practically absorbed or eliminated all other wider used modeling languages, with the exception of Matlab/Simulink, which we do not see as a proper modeling language but as a graphical programming language. UML is quite large and still suffers from the multiple options and interpretation possibilities that, due to its various fields of application, cannot be clarified that easily. Instead, it might be better to create a more explicit variability model for syntactical, methodical, and semantic differences and to configure UML for single projects by suitable selection [Grö10].

The programming language Java has prevailed even more successfully as the primary web and business system language, as well as a teaching language for computer science students.

Therefore, in this as well as the second book "Agile Modeling with UML" UML and Java are consolidated, moderately supplemented and enhanced to allow smooth and integrated use. UML is available in version 2.3 and Java in version 6. UML/P introduced in this book represents a relatively independent and adapted version, a so-called profile of UML, but this profile has been adjusted in some parts by modifications from UML 1.4 to UML 2.3. Because we use Java as the target of generation and test activities, it is certainly of interest to refer to new concepts in Java such as the generics and the assert statement.

Despite or maybe particularly because of the success of both approaches, the gap between the worlds of the model-based software development with UML and agile methods has not really decreased. While agile methods definitely prefer to generate

[2]Translated from the Preface of the German Edition.

code instead of writing it manually, many developers regard the hurdle to successful generation to remain relatively high. Often, the reason for this is the inconvenient and the heavyweight character of the generation process and the relatively high initial effort required to introduce generation tools into the development process. This gap still needs to be closed.

A number of people have directly or indirectly contributed to the creation of the first, and the revision to the second, version of this book. My particular thanks go to Manfred Broy, whose support made this book possible. I would also like to thank my employees and students, especially Christian Berger, Marita Breuer, Angelika Fleck, Hans Grönniger, Sylvia Gunder, Tim Gülke, Arne Haber, Christoph Herrmann, Roland Hildebrandt, Holger Krahn, Thomas Kurpick, Markus Look, Shahar Maoz, Philip Martzok, Antonio Navarro Pérez, Class Pinkernell, Dirk Reiss, Holger Rendel, Jan Oliver Ringert, Martin Schindler, Mark Stein, Christopher Vogt, Galina Volkova, Steven Völkel, and Ingo Weisenmöller who used this book as a basis for their work or who helped to supplement and improve it for the second edition. I would like to thank the former Bavarian Minister for Science, Research, and the Arts, Hans Zehetmair, for the habilitation scholarship award and my appreciated colleague and predecessor Prof. Dr. -Ing. Manfred Nagl for his benevolent support in establishing the chair at Aachen.

My sincere thanks are due to my friends and colleagues, my scientific staff, and the students from Munich for constructive discussions, collaboration in the application examples and reviews of intermediate results of this book in its first edition: Samer Alhunaty, Hubert Baumeister, Markus Boger, Peter Braun, Maria Victoria Cengarle, David Cruz da Bettencourt, Ljiljana Döhring, Jutta Eckstein, Andreas Günzler, Franz Huber, Jan Jürjens, Ingolf Krüger, Konstantin Kukushkin, Britta Liebscher, Barbara Paech, Jan Philipps, Markus Pister, Gerhard Popp, Alexander Pretschner, Mattias Rahlf, Andreas Rausch, Stefan Rumpe, Robert Sandner, Bernhard Schätz, Markus Wenzel, Guido Wimmel, and Alexander Wisspeintner.

Aachen, Germany Bernhard Rumpe
June 2011

Preface to the English Edition

Colleagues have asked when the English version of this book would be published. Finally, here it is. I wish all the readers, students, teachers, and developers fun and inspiration for their work.

I would like to thank all the people that helped me translating and quality checking this book, namely Sabine Blumensath, Robert Eikermann, Timo Greifenberg, Julia Gunder, Sylvia Gunder, Arne Haber, Robert Heim, Lars Hermerschmidt, Gabi Heuschen, Katrin Hölldobler, Andreas Horst, Steffi Kaiser, Carsten Kolassa, Thomas Kurpick, Achim Lindt, Markus Look, Klaus Müller, Antonio Navarro Pérez, Pedram Mir Seyed Nazari, Dimitri Plotnikov, Alexander Roth, Christoph Schulze, Michael von Wenckstern, and Andreas Wortmann.

Aachen, Germany
February 2016

Bernhard Rumpe

Contents