Preface

A common problem in recent years is the growth of software development complexity due to customer demand for more features and fewer errors. Furthermore, due to recent advancements in technology, it has become necessary to utilize software in multiple domains and professional areas. This leads to problems such as development teams becoming experts in one particular area, necessitating an adjustment period when the team starts new projects in other professional areas.

However, software engineering continually offers new tools that, when properly used, can help in the difficult task of developing software complying with the triple constraint of project management (scope, time, and cost) that is cited in numerous sources. Thus, a relatively new software development approach called Model-Driven Engineering (MDE) has appeared.

MDE is an important and emerging approach in software engineering to increase the level of abstraction of the development tasks. In recent years, Model-Driven Engineering has become a critical area of study, as companies and research institutions have started to emphasize the importance of using model as first-class artifacts in the software development process of complex systems.

The mission of this book is to bring researchers, practitioners, and students to one of the most promising fields in computer science, helping all to understand the current state of the art and to know what the future will bring. The objectives include:

- Bringing together the most relevant research on model-driven engineering.
- Updating the scientific literature on model-driven engineering.
- Identifying and address the complexities encountered in the application of model-driven engineering technologies.
- Identify the most important issues to be addressed by scientists in the coming years.

The target audience of this book will be composed of professionals and researchers working in the field of model-driven software engineering in various disciplines. Moreover, the book will also be a reference for researchers, professionals, and students in computer science and related fields. The book will provide a much needed reference on the state-of-the-art of advanced tools and techniques that are either available or under development to support the maximization of the efficacy and efficiency of model-driven software development. It will also provide foundations to professionals, researchers, and academics on the underlying theory and current applications for use in the future advancement of the existing body of knowledge. This combination of theory, applications, and success stories will provide the reader with an important and detailed view of recent developments in the field and lay the background for future research.

Regarding the distribution of chapters, they are distributed as follows:

- Chapter 1 focuses on the implications of the DO-178C standard for model transformations, which are a central part of Model-Driven Engineering approaches.
• Chapter 2 focuses on Software Performance Engineering (SPE), a sub-area of software engineering. In particular, in SPE based on models developed using the Unified Modeling Language (UML).
• Chapter 3 focuses on requirements engineering, a process of constantly changing intentions, goals, and system models, proposing a new method that exploits similarities in semantics of specifications and executable models.
• Chapter 4 focuses on the evaluation of possible parallels to know whether or not the time is ripe for a change of direction in model language development for software engineering.
• Chapter 5 focuses on the development of a distributed temporal meta-data framework for Enterprise Information Systems (EIS) applications to overcome traditional solutions issues.
• Chapter 6 focuses on the production of crop models for the ITK firm. For that, a Domain-Specific Modeling Language has been formalized by a metamodel to create a Crop Model Factory (CMF) prototype.
• Chapter 7 focuses on an e-Commerce metamodel for reuse and interoperability through the use of the Model-Driven Architecture (MDA) standard proposed by the Object Management Group (OMG).
• Chapter 8 focuses on a process, which enables the software architect to validate the system architecture against the architecture-relevant requirements in the development of complex systems.
• Chapter 9 focuses on some issues that hamper widespread adoption of Domain-Specific Languages (DSLs) with particular emphasis on the vagueness of the term DSL and the difficulty of developing DSLs.
• Chapter 10 focuses on modeling notations for expressing various aspects of modern organizations (Enterprise Architecture) with a view to reducing complexity, increasing technology independence, and supporting analysis.
• Chapter 11 focuses on platforms for e-government transactions and suggests a rule-based Domain-Specific Modeling Environment for public services and process integration, avoiding the lack of interoperability of traditional methods.
• Chapter 12 focuses on modernization of legacy systems, intended to provide support for transforming an existing software system to a new one that satisfies new demands. This is done with reverse engineering and models.
• Chapter 13 focuses on the core of the Unified Modeling Language (UML), which is reformalized to avoid ambiguities, contradictions, or redundancies to properly instrument metamodel-based adaptation.

As a conclusion, we think that the book can be used to learn the challenges related to software modeling and new lines of research in which we will work in the coming years regarding the Model-Driven Engineering.

Vicente García Díaz  
University of Oviedo, Spain

Begoña Cristina Pelayo García-Bustelo  
University of Oviedo, Spain

Juan Manuel Cueva Lovelle  
University of Oviedo, Spain

Oscar Sanjuán Martínez  
University of Carlos III, Spain