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.

This book focuses primarily on improvements for software development, supported by two of its main pillars, Model-Driven Engineering (MDE), based on the use of models as a fundamental support in software development, and the specific development of software for mobile devices, one of the areas with the biggest and most exciting changes in recent years.

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

- Bringing together the most relevant research on systems and software engineering.
- Updating the scientific literature on systems and software engineering.
- Identifying and addressing the complexities encountered in the application of new software engineering technologies.
- Understanding 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 any field of systems and software engineering. 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 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 foundation for future research.

Regarding the distribution of sections and chapters, they are distributed as follows:
Section 1 is related to the software development process or software development life-cycle in general. There are several models for such process, each describing approaches to a variety of tasks or activities that take place during it. The eight chapters, though different, are primarily concerned with close aspects to the use of models as a means to facilitate the development of software. Topics covered include security, software vulnerabilities, cost estimation, business services elicitation, metrics, and software process methods.

- Chapter 1 focuses on security goals and software vulnerabilities by the development of a modeling language that can be used in place of four existing languages: attack trees, vulnerability cause graphs, security activity graphs, and security goal indicator trees. Artifacts in the new language can be transformed to and from earlier languages, and a precise definition of model semantics enables an even wider range of applications, such as testing and static analysis.

- Chapter 2 focuses on a security engineering process based on UML security problem frames and concretized UML security problem frames. Both kinds of frames constitute patterns for analyzing security problems and associated solution approaches. The authors describe step-by-step how the pattern system can be used to analyze a given security problem and how solution approaches can be found.

- Chapter 3 focuses on how a framework of modern statistical, computational, and visualization techniques can be used to evaluate and compare the effect of missing data techniques on the accuracy of cost estimation models. For the illustration of the framework, the authors use five missing data techniques: Multinomial Logistic Regression, Listwise Deletion, Mean Imputation, Expectation Maximization, and Regression Imputation.

- Chapter 4 focuses on an approach to elicit business services, and further software services, using the Resources, Events, Agents (REA) business model as the starting point. The proposal aims to perform a value-explorative analysis and modeling of business services at the top level, and model transformations using UML 2 to the system level, by utilizing well-defined mappings.

- Chapter 5 focuses on presenting a study of the basic parameters for estimating the potential in-frastructure and software costs deriving from building and deploying applications on cloud and on-premise assets. Estimated user demand and desired quality attributes related to an application are also addressed in this chapter as they are aspects of the decision problem that also influence the choice between cloud and in-house solutions.

- Chapter 6 focuses on presenting the analysis of 10 recently proposed object-oriented metrics based on cognitive weights. Cognitive weight is the representation of the understandability of the piece of software that evaluates the difficulty experienced in comprehending and/or performing the piece of software. This chapter presents a critical review of existing object-oriented cognitive complexity measures. In addition, a comparative study based on some selected attributes is presented.

- Chapter 7 focuses on introducing the Qualitative Service Elicitation (QSE) method. It applies qualitative research procedures in service elicitation. Service engineering practice lacks lightweight methods to identify service candidates in projects with tight schedules. The QSE provides a systematic method to analyze requirement material in service-oriented systems development with feasible effort by utilizing the procedures of the grounded theory research method to elicit service candidates from business process and use case descriptions.
• Chapter 8 focuses on discussing the rapid growth in technology and the dynamism it presents in our society today, which poses a lot of problems for the software engineering practitioners. The result is a series of software development process methods that can be used to combat or meet up with the problems it creates for developers. On this basis lays the need to develop the model proposed in this chapter to meet up with variations that exist as a result of technological advancement.

Section 2 is related to Model-Driven Engineering, which 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 models as first-class artifacts in the software development process of complex systems. The 11 chapters include topics such as simulation, quality, tools, rule-based environments, requirements, productivity analysis, semantics, legacy systems, domain-specific languages, formalisms, customizations, and transformations.

• Chapter 9 focuses on modern organizations, which need to address increasingly complex challenges including how to represent and maintain their business goals using technologies and IT platforms that change on a regular basis. This chapter reviews the current Enterprise Architecture (EA) modelling landscape and proposes a simple language for the practical support of EA simulation including business alignment in terms of executing a collection of goals against prototype execution.

• Chapter 10 focuses on presenting an extended version of a quality-driven, model-driven approach for database system development. The extension consists of considering the relationship between successive models. In particular, it gives rise to the introduction of a domain ontology as a model preceding the computer-independent model as well as allows one to assess to what extent the successive model is conformant with the preceding model.

• Chapter 11 focuses on proposing a rule-based domain-specific modeling environment for public services and process integration formed by common public service elements and a set of process integration rules. This approach provides a mechanism to integrate the conforming pieces of public transactions among different platforms. In addition, a service and a process meta-model is proposed in order to formalize the information structures needed for their integration.

• Chapter 12 focuses on presenting an approach to software development where model-driven development and software reuse facilities are combined in a natural way. The proposed solution is based on requirements written in semiformal requirements language. This model-based language lets one capture the essence of any software system in the form of domain knowledge and its usage in use case scenarios by wiki-like hyperlinks.

• Chapter 13 focuses on describing how the authors conducted quantitative productivity analysis for a domain-specific modeling language. The analysis shows (1) the significant quantitative productivity gain achieved, (2) the improvement achieved when using an interpreter to automatically generate implementation artifacts as compared to alternative methods when configuring application entities, and (3) the viability of quantitative productivity metrics.
• Chapter 14 focuses on Executable UML, which provides semantics for a subset of actions sufficient for computational completeness. This chapter uses Alf, as the fUML-based action language to describe the operations for iCOMPONENT: the proposed solution for a platform-independent component model for dynamic execution environments. Moreover, a UML profile for modeling components is defined and applied to the development of service-oriented components for dynamic execution environments.

• Chapter 15 focuses on Internet of Things (IoT) as a paradigm that promotes a world in which smart objects and electronic devices are coordinated autonomously to perform a wide range of tasks. The authors define a domain-specific language that allows specifying the coordination and communication between different types of smart objects, regardless of the smart object technical characteristics. The proposed language has been designed to be used in an intuitive and easy way for people without technical knowledge.

• Chapter 16 focuses on modernization of legacy systems. For many years, traditional reengineering has been a solution to software modernization. However, reengineering often fails because it involves ad hoc and non-standardized processes. Software modernization requires technical frameworks for information integration and tool interoperability that allow managing new platform technologies, design techniques, and processes. To meet these demands, Architecture-Driven Modernization (ADM) has emerged.

• Chapter 17 focuses on discussing the role of modeling as a treatment for software engineering since the question about the sufficiency of modeling in software development is still open. The authors believe that software development, in general, and modeling, in particular, based on mathematical formalism in all its stages together with the implemented principle of architectural separation of concerns can become an important part of software engineering in its real sense.

• Chapter 18 focuses on the customization of Enterprise Information Systems (EIS) scale applications, which can be very expensive, incurring substantial additional lifecycle costs to produce and maintain customizations. The development of a temporal meta-data framework for EIS applications seeks to greatly minimize these issues, with the application logic model supporting the capability for end users to define their own supplemental or alternate application logic meta-data.

• Chapter 19 focuses on a review of requirements of the DO-178C standard for Model-Based Development (MBD) and the identification of ways in which MBD can be combined with formal verification to achieve DO-178C requirements for traceability and verifiability of models. In particular, the authors consider the implications for model transformations, which are a central part of MBD approaches, and identify how transformations can be verified using formal methods tools.

• Chapter 20 focuses on e-commerce, which has achieved rapid evolution from simple and static systems on the Web, which provided information and promoted products, to complex systems and dynamic applications that support business processes. The reuse and interoperability are strategies to face the challenge in a dynamic context with rapid technological changes. To achieve these strategies, the authors work with conceptual models that faithfully collect business semantics.
Section 3 is related to Mobile Software Engineering, one of the trendiest areas related to software development. The eight chapters deal with different aspects of mobile software engineering. Topics include augmented environments, malicious applications, real-time services, intelligent environments, usability, mobile software agents, patterns, distributed context management, and quality attributes.

- Chapter 21 focuses on a real augmented environment and its associated mobile interactions based on wearable computers with appropriate interaction devices that can be either classical computer-interaction devices or real objects augmented with computer interfaces called tangible objects. After presenting the main principles, the authors describe a concrete platform, related models, formalisms, and the development processes, as well as several applications.

- Chapter 22 focuses on an adequate security support on mobile devices to avoid malicious applications that damage the device or perform unauthorized accesses to personal data (such as personal contacts or business mails). This chapter describes the security support of the current commercial mobile devices along with a set of approaches that have been proposed in the scientific literature to enhance the security of mobile applications.

- Chapter 23 focuses on system architectures delivering real-time services to customers, which must be flexible, scalable, and support a wide range of communication channels. This chapter presents an architecture that was designed to support multiple delivery channels and was successfully used to implement mobile banking services. The considerations behind the design and the approach used to deliver SMS-based mobile services using service-oriented architecture principles are reviewed.

- Chapter 24 focuses on pervasive intelligent environments, where Wireless Sensor Networks (WSNs) surround and serve people at any place and any time. A proper usability is considered essential for WSNs supporting real life applications. With this chapter, the authors aim at ease of use for specifying new applications that have to autonomously cope with expected and unexpected heterogeneity, sudden failures, and energy efficiency.

- Chapter 25 focuses on mobile software agents, which need the presence of a middleware to allow them to exist and provide them the means to develop their potential. There exist a number of such platforms with different features, but also some limitations that must be taken into account if they are going to be used in the development of mobile applications. In this chapter, the authors discuss in detail related questions and survey the most popular agent platforms to find out how they can be used in mobile environments.

- Chapter 26 focuses on a novel generic architectural model approach for organizing patterns as a solution for efficient utilization of patterns in a specific domain. In this approach, the identification of relevant patterns is considered as the process of reducing the set of candidate patterns by domain-implied constraints, such as functionality, platform, and problems. These constraints can be incorporated in a domain-specific generic architectural model that reflects the commonalities in the solutions of the particular domain.

- Chapter 27 focuses on the convergence between mobile telecommunications and the Future Internet. Addressing the advanced requirements of self-improving Personal Smart Spaces (PSSs) regarding the establishment of a robust distributed context management framework is a challenging task. This chapter elaborates on a context modelling and management approach that is suitable for addressing the PSS requirements and provides experimental evaluation evidence regarding its performance.
Chapter 28 focuses on discussing specific quality attributes with respect to device-side software architectures providing multimedia and sensor capabilities. This chapter focuses specifically on device-side client architectures, rather than network or server architectures. Specific domain requirements and quality attributes are first derived through a synthesis of current research and industry trends, and subsequently analyzed.

As a conclusion, we think that the book can be used to understand new research on innovations in systems and software engineering, which will be an exciting area of work in the coming years in the software development industry.

Vicente García Díaz  
University of Oviedo, Spain

Juan Manuel Cueva Lovelle  
University of Oviedo, Spain

B. Cristina Pelayo García-Bustelo  
University of Oviedo, Spain