Preface

The idea behind this book came during a group meeting in a European R&D project. We came to a realization that while there exists a plethora of research articles that detail activities related to embedded systems design, they usually do not show the whole picture. The academic research articles are more visible as compared to design activities and practices present in the embedded systems industry, which are not often brought to light, either due to confidentiality reasons or because the results are usually presented in industrial exhibitions and conferences, which may not be accessible to academia. It is for this reason that we decided to go ahead with the idea of editing a book that targets not only current academic practices but also relevant industrial trends for embedded systems design in order to provide valuable insight to embedded systems designers, developers, and students about the whole spectrum of embedded systems design covering various topics such as model-based methodologies, software optimisation techniques, hardware implementation strategies, and general design choices that result in existing embedded systems.

BACKGROUND

Real-Time Embedded Systems (RTES) are omnipresent in our personal and professional daily lives: from physical/environmental monitoring via Wireless Sensor Networks in the field of telecommunications to the domains of medicine, defense, avionics, and transport; it is difficult to find a place where RTES have not made their mark.

Low power and energy consumption factors have become critical to the design and implementation of these systems in order to ensure maximum battery life, while providing adequate balance between performance and system lifetime. According to the ITRS roadmap, “power defines performance” and is the single most important factor for the design of future RTES (ITRS, 2011).

In addition to these critical factors, rapid evolution and continuous technological advances in RTES, along with a sharp increase in targeted application domains, have led to new challenges in the specification, design, and implementation of these systems, like increased development life cycles, non-recurring engineering costs, and poor synergism between the different development teams related to hardware and software development, resulting in decreased productivity. Therefore, effective design methodologies and efficient design tools are needed to address the above-mentioned challenges while resolving issues related to system complexity, power consumption, and verification.

It is thus evident that in the near future embedded systems will become more difficult to develop unless new and effective methodologies are developed to combat the above-mentioned issues. New domains such as Cyber-Physical Systems (Foundations for Innovation in Cyber-Physical Systems, 2013) can be
viewed as the next generation of embedded systems that couple the latter with physical processes, along with human-in-the-loop. These complex systems will benefit from effective methodologies targeting real-time and embedded systems.

In recent years, numerous research works have been carried out to reduce the above-mentioned challenges related to RTES: task scheduling for power management, software power optimizations (for example code compression and coding), and low-power communication techniques. Effective Network-on-Chip communication-based architectures have been proposed among other diverse approaches at various levels of abstraction, from the electronic Register Transfer Level (RTL) up to the system design level, to address these issues (Lee, et al., 2006).

It has been observed that a high abstraction-level design methodology produces more effective impacts, due to increased degree of flexibility, as compared to tighter constraints applicable at either transactional-level modeling (via SystemC) or RTL (Cai & Gajski, 2003). Hence, high-abstraction-based system design approaches such as Model-Based Design (MBD) or Model-Driven Development (MDE) (Selic, 2003) have been developed, which enable system specification via graphical notations or languages like UML (Unified Modeling Language) (OMG, 2011a) to reduce system complexity, and to effectively partition the system design according to the classical Y-chart hardware/software co-design paradigm (Gajski & Kuhn, 1983). Additionally, MDE integrates different tools, technologies (like the Eclipse Modeling Framework or EMF), and standards (UML and related profiles for high-level system specifications, such as SysML [OMG, 2012] and MARTE [OMG, 2011b]). Finally, Model Transformations can automatically generate executable models or code from these high-level design models (Sendall & Kozaczynski, 2003).

High-abstraction model-based design methodologies have some crucial advantages. The abstraction achieved by a modeling approach permits one to abstractly emphasize the overall system structure, while bypassing details related to implementation and associated technologies. This enables conception and development of huge complex systems in a speedy and efficient manner. These approaches enable representing a system or a part of the system with different points of view, which permit system separation by aspects related to specific domain views. These approaches can also be found in Aspect-Oriented Programming (AOP) and Domain-Specific Languages (DSLs) (Mernik, et al., 2005).

The European research initiative has also taken up the challenge for real-time and embedded development processes, and many research efforts have been carried out during the last few years in many European-funded R&D projects (such as EU FP7, ARTEMIS, and EU members’ national projects) related to embedded systems, to decrease time-to-market and to address issues related to increasing costs, complexity, and productivity.

This book presents an overview of the results of various R&D projects in order to provide a reference text that addresses system engineering principles for embedded systems to disseminate them in the embedded system developers and user communities.

The book provides contributions from practitioners of the embedded systems community, both from industry as well as academia, involved in European Research projects targeting real-time and embedded systems. These projects focus on a wide spectrum of design principles related to these complex systems, ranging from their specification at high abstraction levels using standards such as UML and related profiles (SySML, MARTE, EAST-ADL [East-ADL Association, 2014], AUTOSAR [2014], etc.) to intermediate design phases related to verification, timing, scheduling, and performance analysis, down to low-level implementation details related to the underlying execution platforms, such as synthesis, placement, and
routing on FPGAs, System-on-Chips (SoCs), and Network-on-Chips (NoCs). In particular, the book covers the research carried out to address specific embedded system constraints related to hard timing constraints, limited memory and power use, predefined hardware platform technology, and hardware costs.

The book provides an overview of different approaches developed out in the embedded systems domain in academia and industry. The topics covered in the book are extremely current as they tackle issues related to increasing design productivity, lowering energy consumption levels, increasing performance, and decreasing the overall time-to-market, all aspects that have been identified as core in the Horizon2020 EU R&D program as well as in the INCOSE 2020 vision for systems engineering.

The book tackles various interesting and relevant aspects: from model-based design (such as requirements management, model transformations and adoption of high-level standards such as EAST-ADL, SysML, and MARTE) resulting in early design space analysis, verification, and automatic code generation for implementation of embedded systems, to topics such as development of low-power embedded systems and emulation techniques, fault tolerance, software development techniques such as aspect-oriented programming, virtualization techniques, to hardware implementation techniques like dynamic reconfiguration of embedded systems to manage system configurations and failures.

**TARGET AUDIENCE**

The target audience of this book are designers of embedded software, academicians, students, practitioners, professionals, and researchers working in the field of real-time and embedded systems development. This book is a rich source of information covering theoretical analysis, algorithms, and practical applications of software engineering for real-time and embedded systems.

The book will also be of great advantage to a reader interested in gaining a basic understanding of embedded systems design perspectives in research and industry and at which level the current standardization efforts in the field have been used. This could be, for example, the design and development manager in an organization, a system architect, or a researcher interested in learning the current embedded system design perspectives in both industry and academia.

**CHAPTER OUTLINES**

The chapters of the book are organized in three sections. The first focuses on design techniques addressing specific issues in the development of embedded systems; they range from simulation of early system models to optimization of hardware components. The second includes chapters presenting complete development methodologies that have been defined as the result of collaborations among academic and industrial partners in various projects. The third comprises chapters providing a more industrial and application-oriented view on the development of embedded systems.

**Section 1: Research Perspective – Software and Hardware Design**

The first three chapters of Section 1 deal with modeling and verification in the broad sense of the term of the control logic in embedded and cyber-physical systems.
In Chapter 1, “Collaborative Development of Dependable Cyber-Physical Systems by Co-Modeling and Co-Simulation,” Fitzgerald, Pierce, and Larsen explore the challenges related to the simulation of complex cyber-physical systems and propose a method developed in the European project DESTECS for the simulation of heterogeneous systems through the combination of different tools for different parts of the system.

In Chapter 2, “UML MARTE Time Model and its Clock Constraint Specification Language,” Mallet, Peraldi-Frati, Deantoni, and de Simone present an approach to the modeling and analysis of timing constraints in real-time embedded systems based on the UML MARTE profile and on its notion of clocks.

In Chapter 3, “Symbolic-Based Monitoring for Embedded Applications,” Mouttappa, Maag and Cavalli introduce a formal approach, which was applied in the DIAMONDS European project, for checking the conformance of execution traces of an implemented embedded system with respect to its specification.

The final four chapters of Section 1 cover design issues that are closer to the hardware platform level and address hardware-related metrics and properties, such as energy dissipation or fault tolerance.

Chapter 4, “Designing Resource-Constrained Embedded Heterogeneous Systems to Cope with Variability” by Gray, Acquaviva, and Audsley, considers the upcoming problems resulting from the performance and energy dissipation variability introduced by the latest silicon manufacturing processes at the nano-dimensions. It proposes a model-based flow that facilitates the creation of a runtime environment that is aware of the variability effects and that is able to mitigate those effects by remapping computation to different processors, parallelizing operations, and optimizing the energy dissipation by dynamically scaling voltage.

In Chapter 5, “Vulnerabilities of Secure and Reliable Low-Power Embedded Systems and their Analysis Methods: A Comprehensive Study,” by Druml, Menghin, Steger, Krieg, Genser, Haig, Bock, and Grinschgl, a number of power analysis and minimisation techniques are discussed, followed by a review of security threats that can explore power-related vulnerabilities, such as power profile side channel attacks and fault injection attacks.

Chapter 6, “An Aspect-Oriented Approach to Hardware Fault Tolerance for Embedded Systems” by de Andrés, Ruiz, Espinosa, and Gil, presents a meta-programming approach to extend an HDL-based design flow aiming to facilitate the optimization and evaluation of fault-tolerance and security features in hardware IP cores.

In Chapter 7, “Optimized System Level Design Methods for NoC-Based Many Core Embedded Systems,” Ying, Hofmann, and Hollstein present a methodology and respective tool flow to optimise performance and energy dissipation of the 3D XHiNoC network-on-chip. It does that by relying heavily on search-based heuristics to tune task mapping, core placement, network topology, and routing.

Section 2: Development Methodologies and Tool Suites

The second section of the book includes chapters that look at several phases and activities in the development of embedded systems, either by defining methodologies that span over them, or by describing tool suites that support them.

In Chapter 8, “MADES FP7 EU Project: Effective High Level SysML/MARTE Methodology for Real-Time and Embedded Avionics Systems,” Bagnato, Quadri, Brosse, Sadovykh, Soares Indrusiak, Paige, Audsley, Gray, Kolovos, Matragkas, Rossi, Baresi, Crippa, Genolini, Hansen, and Meisel-Blohm present the model-driven approach to the development of embedded systems that was the outcome of
the MADES European project; the approach covers many aspects, from high-level modeling, verification and simulation, to the allocation of functions on the target hardware, to the automated generation of code tailored to the target platform.

In Chapter 9, “Requirements Refinement and Component Reuse: The FoReVer Contract-Based Approach,” Baracchi, Cimatti, Garcia, Mazzini, Puri, and Tonetta describe the mechanisms developed in the FoReVer project, funded by the European Space Agency, to develop complex systems in a modular, component-based manner through the notion of “contracts” among modules; the proposed approach allows designers to formally verify the correctness of the decomposition of requirements when moving through the different development phases of a space system and in particular, from system-level to software requirements.

In Chapter 10, “Model-Based Analysis and Engineering of Automotive Architectures with EAST-ADL,” Tucci-Piergiovanni, Chen, Mraidha, Lönn, Mahmud, Reiser, Kolagari, Yakymets, Librino, Torchiaro, and Lanusse introduce the model-based development approach of automotive systems based on EAST-ADL defined in the MAENAD European project; in particular, the chapter describes how different analysis techniques (for studying functional, safety and timing properties of the system, and for its optimization) are integrated and automated in the MAENAD approach.

In Chapter 11, “Fostering Analysis from Industrial Embedded Systems Modeling,” Bourdellès, Li, Quadri, Brosse, Sadovychk, Gaudin, Mallet, Goknil, George, and Kreku present the tool suite developed in the PRESTO ARTEMIS project for the validation of non-functional properties of embedded systems from the early phases of development; the tool suite allows developers to generate execution traces from instrumented code, specify the functional and non-functional properties of interest, and then verify that the generated traces have the desired properties.

In Chapter 12, “A Model Driven Engineering Method for DRE Defense Systems Performance Analysis and Prediction,” Falkner, Chiprianov, Falkner, Szabo, and Puddy describe a tool suite that allows embedded systems developers to analyze and predict the performances of the designed system from the early development phases.

**Section 3: Industry Perspective and Applications**

The third section of the book includes chapters that look at embedded systems on one hand from the point of view of the applications and on the other hand from the point of view of industries involved in the field.

In Chapter 13, “Industrial Applications of Emulation Techniques for the Early Evaluation of Secure Low-Power Embedded Systems,” Druml, Menghin, Steger, Krieg, Genser, Haid, Bock, and Grinschgl build on the results presented in Chapter 5 and apply them to evaluate and verify designs of contactless smart cards.

In Chapter 14, “Dynamically Reconfigurable Embedded Architectures for Safe Transportation Systems,” Harb, Niar, and Saghir show how the features of modern FPGAs, and in particular Dynamic Partial Reconfiguration, can be leveraged to tailor the functions offered by a Driver Assistant System (DAS) according to different situations, for example to increase the accuracy of obstacle tracking algorithms when the density of obstacles increases (e.g., during rush hour).

In Chapter 15, “Embedded Virtualization Techniques for Automotive Infotainment Applications,” Violante, Macario, and Campagna study the features of several virtualization technologies that are available for the embedded systems domain; with particular reference to the automotive domain, where they can be used to build infotainment systems that include functions requiring different OSs. The chapter shows the benefits and drawbacks of each possible solution.
In Chapter 16, “Studying Individualized Transit Indicators using a New Low-Cost Information System,” Castillo, Fernández-Ares, García-Fernández, García-Sánchez, Arena, Mora, Rivas, Asensio, Romero, and Merelo describe the realization of a low-cost embedded system that exploits the Bluetooth devices that are available on modern cars to track the passages of vehicles at specific locations around a region; the system has been deployed around the city of Granada, and the chapter describes some statistics retrieved through it.

In Chapter 17, “Mission Critical Embedded System Development Process: An Industry Perspective,” Genolini and Crippa summarize their experiences gained in several decades of work on development of embedded systems; they outline key critical aspects and offer some lessons learned.

In Chapter 18, “Framework-Based Debugging for Embedded Systems,” Tanyeri, Messiter and Beckett illustrate, through the use of an industrial tool suite, a modular and open approach to the problem of debugging embedded systems that rely on several different OSs and platforms; the approach hides the details of the underlying hardware and operating system, thus facilitating the debugging activities of the development team.

In Chapter 19, “Industrial Experiments in IMS, ATC, and SDR projects of Property Verification Techniques,” Gaudin explores the issues concerning the verification, in an industrial setting, of properties of interest of a system under development; in particular, it outlines the application of verification techniques to several industrial case studies.

**CONCLUSION**

In conclusion, this book is interesting for current and future embedded systems designers and developers. It covers a wide spectrum of issues related to embedded systems design ranging from high-level methodologies, software development techniques, and hardware implementation on execution platforms. The book not only summarizes current results in the design of embedded systems, but, also sheds light on the future of such systems, towards so-called cyber-physical systems. The book can thus provides insight on current practices, related issues, and their solutions, and how these solutions are giving rise to the next evolutionary trend for embedded systems design.

_Alessandra Bagnato_
_Softeam R&D, France_

_Leandro Soares Indrusiak_
_University of York, UK_

_Imran Rafiq Quadri_
_Softeam R&D, France_

_Matteo Rossi_
_Politecnico di Milano, Italy_
REFERENCES


