Preface

The embedded systems market is a lively—some would say volatile—place. There is a growing demand for products that make the best use of rapidly improving computer hardware to create everything from game consoles to flight controllers. In this setting, the developers of embedded systems have to form creative teams out of disparate engineering disciplines. For example, a product design team might encompass software, mechanical, electrical and control engineers. However, effective collaborative design is not simply a matter of sharing a whiteboard with each other—the bases of engineering disciplines are different, and perhaps the biggest gulf is between software and control engineering. Control engineers describe how phenomena evolve and flow over continuous time, but software is described using logic to relate discrete events.

The semantic gaps between engineering disciplines cost time and money because the results of misunderstandings are often only detected when the physical product is built and software fails to control it properly. Traditional product development involves specialist engineering groups working independently on aspects of the design which is passed between them, sometimes being misinterpreted and distorted as it goes. This is a particularly pressing problem when we attempt to design for resilience: dependability cannot be sprinkled over a completed design, but needs to be integrated from the outset. Indeed, our experience suggests that around 80 % of embedded software is related to complex supervisory control which includes switching between modes or dealing with error detection and recovery. It is essential, then, to have methods and tools to help manage the risk of early-stage design flaws.

This book is a response to the challenge of delivering effective multi-disciplinary design. It builds on the premise that early analysis of design models could lead to early detection of errors and performance bottlenecks, and the models themselves can serve as common bases for communication and the exploration of design alternatives. But how can model-based methods work if the engineering disciplines describe aspects of the product and its environment in such different ways? We focus on the creation and analysis of collaborative models ("co-models"), composed of discrete-event and continuous-time models. Typically, these contain discrete-event models of control elements to be realised on computers, coupled with continuous-time models of controlled plants and the physical environment. However, rather than forcing diverse disciplines into a single common modelling framework, we show how it is possible to link otherwise separate tools for discrete-event and continuous-time modelling through a harness that allows them to share data during simulations to create a unified "co-simulation". Co-simulation between tools supports collaboration between designers, allowing engineers from each discipline to continue work within familiar formalisms while being able readily to judge the effects of a design decision in one domain on the behaviour of the other. For example, we might choose to resolve a known hardware fault either by using different sensors or by more complex control software. With co-simulation, we can trade off these alternatives on such factors as performance, energy consumption and cost before a commitment is made.

The methods and tools for co-modelling described in this book were developed in a collaborative research project, "Design Support and Tooling for Embedded Control Software" (DESTECS).[1] We were fortunate in DESTECS to have the cooperation of several pioneering companies, who endured the instabilities of our nascent methods and tools, applying them in several domains. Their experience demonstrated the value of co-modelling in reducing design iterations, easing the development of sophisticated software control and supporting dependability. We hope that this book gives the reader a sense of the potential for innovation enabled by methods and tools that support technically well-founded collaboration across discipline boundaries.

Structure of the Text

Our goal is to present methods and tools for co-modelling, co-simulation and design space exploration in a thoroughly practical way, with running examples. The book is structured in three parts:

1. Part I introduces the technical basis of co-modelling and co-simulation using one Continuous-Time (CT) and one Discrete-Event (DE) formalism. Chapter 1 describes the need for collaborative design and the challenges in providing methods and tools to support it. We then introduce core concepts that underpin the rest of the book (Chap. 2). In Chaps. 3 and 4, we present the specific CT and DE technologies that we propose to link through co-modelling: respectively, bond graphs, supported by 20-sim, and the Vienna Development Method (VDM). Both are comprehensive formalisms, so in these chapters, we aim to give the reader a sense of the main features of each. We then introduce the Crescendo technology for co-simulation (Chap. 5). Finally, Chap. 6 discusses the use of structuring mechanisms to promote the reuse of controller models.

[1] European Union Framework 7 project CNECT-ICT-248134, January 2010–December 2012 (see http://www.destecs.org/).

2. In Part II, we move from foundations to the application of co-modelling. Chapter 7 introduces two case studies (a line-following robot and a ChessWay personal transporter), which are used in this part of the book. The process by which a co-model is developed is discussed in Chap. 8. Chapter 9 introduces techniques for co-modelling faults and fault tolerance mechanisms, while Chap. 10 examines the support for exploring large design spaces in the search for optimal solutions. Chapter 11 brings these strands together, describing how the technology has been applied on other industrial applications.

3. Part III considers more advanced topics. Chapter 12 reports the experiences of three industry users following their experimental deployment of the Crescendo technology. Chapter 13 gives a technical discussion of the semantics of the co-simulation framework underlying the Crescendo technology and explains how it can be re-used with other CT and DE technologies. Finally, Chap. 14 positions our work in the broader setting of model-based collaborative design and sets out the challenges posed by the development of cyber-physical systems.

4. The appendices include summaries of VDM and 20-sim, a catalogue of design patterns for co-models, and an abstract VDM model of the ChessWay with a focus on its safety aspects. A list of acronyms and a glossary of the main terms used in this book are included.

Using the Book

The book is aimed at both researchers and practitioners in embedded systems development. Among researchers, the book should be of interest to those working in cyber-physical systems, embedded systems design and formal methods; in control engineering, the material should be of interest to those working on advanced control and modelling technology, especially for fault-tolerant and resilient systems. Among practitioners, we target the book at those in research and product development with an interest in improved design processes and tools. Among academics, we expect the text to be of value for those teaching embedded software development at all levels. We recommend that all readers make use of the Crescendo tool for hands-on experience and access the additional content, including tutorials and training material, on the accompanying web site (see below).

In keeping with the spirit of co-modelling, we assume that the reader has some experience in either software development or control, but we do not assume both. In Part I of the book, the introductions to CT modelling in Chap. 3 and DE (computing) modelling in Chap. 4 are written with readers from both backgrounds in mind.

Practitioners with an interest in the techniques of co-modelling and co-simulation are invited to approach the contents of Parts I and II in the order presented. Most of the chapters in the technical flow of the book assume that preceding chapters will have been covered. The advanced topics in Part III are relatively independent of one another. Among the advanced topics, Chap. 13 will be of technical interest mainly to those working on the formal semantics of modelling languages and so may be omitted by others on a first reading.

Readers with a primary interest in engineering management may wish to cover the motivation and technical foundations in Chaps. 1 and 2, example case studies in Chap. 7 and elements of co-model creation methodology in Chap. 8, followed by industry applications and deployment experience in Chaps. 11 and 12 and future directions for the technology in Chap. 14.

Accompanying Web Site

The accompanying web site, www.crescendotool.org, provides additional material, including tool support for co-simulation, as described in the book, additional example co-models that can be used with the tool and course material. We invite readers wishing to use the material for teaching or research to take the distribution only from this web site and contact the editors for further support.

Acknowledgments

DESTECS was supported by the European Commission under the Seventh Framework programme. We are grateful to the expert reviewers, Mr. Bernard Dion (Esterel Technologies) and Prof. Reinhard von Hanxleden (Kiel University), for their constructive suggestions and recommendations throughout the project. It is a pleasure to thank the many contributors to the book, particularly Kenneth Pierce, Carl Gamble, Jan Broenink, Job van Amerongen, Christian Kleijn, Augusto Ribeiro, Kenneth Lausdahl, Bert Bos, Sune Wolff and Joey Coleman. We are very grateful to Koenraad Rombaut and Peter van Eijk, who kindly agreed to be interviewed about their industrial application of the technology that we developed, allowing us to recount their experiences in Chap. 12. We gladly acknowledge our other colleagues from DESTECS who contributed to the technology: Claire Ingram, Kim Bjerge, José Antonio Esparza Isasa, Claus Ballegard Nielsen, Xiaochen Zhang, Yunyun Ni, Angelika Mader, Jelena Marinčić, Stefan Groothuis, Peter Visser, Frank Groen, Marcel Groothuis, Dusko Jovanovic, Jan Remijnse, Eelke Visser, Michiel De Paepe, Yoni De Witte, Roeland Van Lembergen, Wouter Vleugels, Kim Visser and Jeffrey Simons. We are grateful to Nick Battle, Stefan Hallerstede, Hiroshi Sako and all those who provided feedback to us on draft material. We would also like to thank Martin Peter Christiansen for providing the T1X tractor example and simulation results. At a personal level, we are deeply grateful to our families and friends for their patience and support since the genesis of this book, especially John Hudson, Margit Sandvang Larsen and Natalie Ree.

Newcastle upon Tyne, UK John Fitzgerald
Aarhus, Denmark Peter Gorm Larsen
Haarlem, The Netherlands Marcel Verhoef

http://www.springer.com/978-3-642-54117-9

Collaborative Design for Embedded Systems
Co-modelling and Co-simulation
Fitzgerald, J.; Larsen, P.G.; Verhoef, M. (Eds.)
2014, XXI, 385 p. 244 illus., 14 illus. in color., Hardcover
ISBN: 978-3-642-54117-9