Preface

This book provides a detailed description of a model-based development approach for Globally-Asynchronous Locally-Synchronous Distributed Embedded Controllers (GALS-DECs) development.

Embedded controllers can be seen as computer systems performing dedicated tasks. They are usually embedded in larger systems, such as industrial machines, vehicles, buildings, home appliances, as well as in some safety-critical systems and medical devices. Most of the time, embedded controllers are forced to exhibit deterministic behavior allied to real-time constraints, offer high performance associated with low power consumption, and are normally constrained by a reduced time-to-market.

Currently, specification of embedded controllers and GALS-DECs is mainly supported by software programming languages and Hardware Description Languages (HDLs), complemented by specific modeling languages. These controllers are commonly implemented in heterogeneous platforms and validated through code simulations and tests.

In complex controllers with millions of possible states, methods relying on simulations and prototype testing are time-consuming tasks and cannot ensure that the controller is free of bugs, as far as it is not possible to cover all possible evolutions and reachable states.

In this sense, model-based development approaches are of special interest to circumvent those weaknesses. These approaches use models not only to improve the level of abstraction of the specification (enhancing the understanding of the behavior of the distributed controller as well as the communication among the stakeholders), but also to provide support to other development phases, namely simulation, verification, implementation using automatic code generators, and final deployment into specific platforms.

This book proposes a model-based development approach that improves the model-based development approaches previously proposed. The new approach aims to avoid a limitation from previous approach, where it was required: (1) to generate the state-space of the global model or (2) to generate the state-space of the reduced model, to make a proper implementation. However, sometimes (1) it is not possible to generate the full state-space (because it is too big) and (2) it is not possible to reduce the model (to a size that enables the state-space generation) or it is not easy to do it. The proposed approach relies on Petri nets extended with a few new concepts, namely time-domains and asynchronous-channels, to support the specification of distributed embedded controllers. Those Petri net models allow the creation of platform- and network-independent models supporting the use of design automation tools. These models support distributed controllers simulation, verification, and implementation, using the cloud-based IOPT-Tools framework, that, among others, includes a simulation tool, a model-checking tool, and automatic code generators, which improve productivity, reducing the development time and eliminating errors from manual coding.

This book is structured in five chapters. The first chapter is devoted to an introduction to the topic and is composed by six sections. Embedded controllers and distributed embedded controllers are defined in the first section. Then, usual development approaches are briefly described. After that, the concept of model-based development is introduced. A list of modeling formalisms is presented in the fourth section, and in the following section the use of Petri nets is justified. Finally, the model-based development approach proposed in this book is introduced.

The second chapter addresses the characterization and comparison with other related works using Petri nets, presenting Petri nets and several extensions that make them suitable to develop embedded controllers and distributed embedded controllers. Petri nets are introduced in the first section and then non-autonomous Petri nets classes are presented. After that, execution semantics, priority concept (used to solve conflicts), boundedness concept, and test arcs are presented. Afterwards, the IOPT (Input-Output Place-Transition) Petri nets class, resulting from the extension of Place-Transition nets with a set of characteristics amenable to describe the interaction of the controllers with the environment, is described. Petri nets classes supporting the specification of GALS (Globally-Asynchronous Locally-Synchronous) systems are also presented, and finally a list of communication channels used in several Petri nets proposals is briefly described.

The third chapter is devoted to the presentation of the proposed model-based development approach and Petri nets extensions to support it. The proposed approach is described in the first section of the chapter, then the Petri nets class in use is extended to support this approach, and finally algorithms to support the verification and the implementation of the created models are described. The proposed approach supports the model-based development of Globally-Asynchronous Locally-Synchronous Distributed Embedded Controllers (GALS-DECs) through platform- and network-independent Petri net models. These models support the documentation, the simulation, the verification, and the implementation. To support this development approach, the Petri nets class in use is extended with a set of concepts, among which are time-domains and asynchronous-channels. Three types of asynchronous-channels are proposed, one covering the simple asynchronous communication between two sub-models, while the other two addressing the status of communication completeness in former asynchronous-channels. An algorithm to support the state-space generation (which supports the verification) and one algorithm to decompose the global model into a set of sub-models (that support the components implementation) are presented. Finally, the meta-model of the Place/Transition nets extended with time-domains and asynchronous-channels is presented.

The fourth chapter is devoted to illustrate the applicability of the approach using an example of a distributed controller targeting a traffic application managing the number of vehicles in a restricted area. The distributed controller is composed by a set of modules, each of them specified using a Petri net model, and interconnected using a set of asynchronous communication channels. Overall, the controller can be seen as a Globally-Asynchronous Locally-Synchronous Distributed Embedded Controller. The development of the distributed embedded controller starts validating each of the components separately, and ending up with the validation of the overall model.

Finally, the fifth chapter presents conclusions and points to some future works. Differences between the model-based development approach proposed in this book and similar model-based development approaches previously proposed are discussed.

Caparica, Portugal — Filipe Moutinho
June 2015 — Luís Gomes

Springer
http://www.springer.com/978-3-319-20821-3

Distributed Embedded Controller Development with Petri Nets
Application to Globally-Asynchronous Locally-Synchronous Systems
Moutinho, F.; Gomes, L.
2016, XII, 79 p. 37 illus., 33 illus. in color., Softcover
ISBN: 978-3-319-20821-3