

Modeling and synthesis of the network in distributed embedded systems

Emad Ebeid

Department of Computer Science, University of Verona

(emadsamuelmalki.ebeid@univr.it)

Design of distributed embedded systems is a novel challenging task and it requires to raise the level of abstraction to overcome the complexity of the design. In particular, modeling languages and custom profiles are necessary to address network specification at this level of the abstraction. In order to verify the NW design model and reach the actual implementation, model manipulation and code generation are needed not only for simulation and performance analysis of the design model but also to refine the model and reduce the level of abstraction. In this work, I propose the use of UML diagrams combined with a formal computational model as a key solution to generate code, solve the NW design problem and manipulate each solution to generate different NW alternatives. This thesis proposes a formal framework to capture the application requirements, library of network components, environment description and the rules to compose them. The framework generates code for design verification by simulation and provides back annotation mechanism of the simulation results to refine the original model. The resulting code is used in the final implementation of the network. Moreover, the framework abstracts existing RTL IPs into UML for component reuse.

This thesis presentation is organized as follows. Section I gives an overview about network synthesis and the problems to be solved, while Section II describes the proposed methodology to solve such problems. Section III reports the progress done as well as the current and future work.

I. CONTEXT AND PROBLEMS

Distributed embedded systems are distributed applications of networked embedded system (NES) which are special-purpose, resource-constrained nodes interacting together through communication protocols to achieve a common goal. They are at the basis of pervasive computing, ambient intelligence as well as Internet-of-Things and bring significant advance in important application fields, e.g., home automation, health care monitoring and industrial plant control.

Design complexity of this kind of embedded system applications requires designers to focus on abstraction levels much higher than the implementation level. In this context, computer-aided design should be fruitfully applied not only to each network node, as currently done, but also to the communication infrastructure among them. Therefore, the design of distributed embedded systems requires to address the following open issues:

- Design of NES needs a common representation of different aspects (HW, SW, computation, communication,

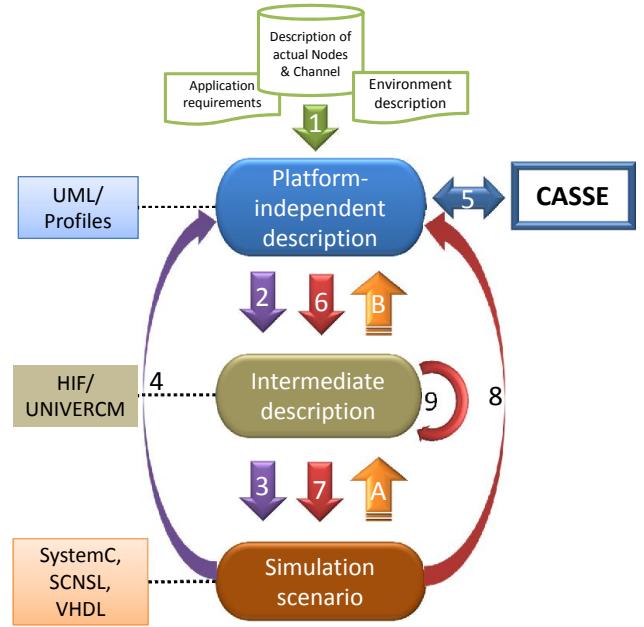


Fig. 1. Design methodology.

analog, and digital) and efficient simulation.

- Methodology should start from a standard representation of the requirements and behaviour of the whole distributed system.
- Automatic design-space exploration and synthesis of the network part of the system (type of channels, protocols and related HW/SW interfaces).
- Simulation and generation of the final system must be used to support analysis and synthesis steps.

In literature, network design for embedded systems has been addressed especially in the context of wireless sensor networks and network-on-chips. In particular, the so-called *network synthesis* problem has been stated in [7], [8]. Synthesis algorithm produces a network implementation that satisfies all end-to-end constraints and that is optimal with respect to some metrics (e.g., cost or power consumption). For instance, the COmmunication Synthesis Infrastructure (COSI) [8] describes and solves a mixed integer linear programming problem in the specific context of building automation without considering the design of each node. The Communication-Aware Specification and Synthesis Environment (CASSE) is a more general approach which describes the problem in terms

of tasks, data flows among them, nodes, abstract channels and environmental constraints. CASSE solution consists in mapping tasks onto nodes, data flows onto abstract channels and positioning nodes in the environment. Both approaches do not rely on a *standard representation* of requirements (from the initial user specification) and do not provide a direct path to the validation of the design through simulation. Thus, a unique, mathematical model and heterogeneous descriptions are necessary to describe network components, in order to allow system integration, manipulation and code generation. Therefore, UML and the Heterogeneous Intermediate Format (HIF) supported by a model of computation named UNIVERCM which captures discrete- and continuous-time aspects of the complex system are used in this thesis.

The goal of this thesis is a design methodology [1] and tools [2], [3] for modeling and synthesis of the packet-based network in distributed embedded systems. The methodology is based on two parts; one part regards the adoption of a common representation framework (e.g., based on UML with custom profiles, SysML and HIF) to describe the different aspects of a distributed embedded system and the other part regards the use of this common representation to synthesize the network and to generate efficient simulations.

II. PROPOSED METHODOLOGY

Figure 1 shows the design flow which starts by using UML to describe the application specification independently from the actual implementation and then goes to an intermediate description for formalization and manipulation of the NW design to generate code for efficient simulation and performance analysis. After each step, results are back annotated to UML for further refinement and documentation. The design specification of the networked embedded application (i.e. application requirements, environment constraints and a description of actual nodes and channels) is extracted and modeled in UML with the support of custom profiles to assign semantics to them (label 1).

A crucial aspect in this step is the functional validation of the model and the estimation of communication requirements (e.g. throughput and latency requirements) for each data flow. For this purpose, the transformation step of UML models to an executable code is necessary. Therefore, transformation tool is developed to extract design information from UML models and represent it by an intermediate representation (2) to generate executable code such as SystemC/TLM or VHDL (3). This step is presented in [2], [3]. The simulation results such as throughput and latency can be generated and back annotated (4) to the original UML model for further refinement.

The refined UML diagram along with an analytical model for network synthesis named CASSE [7] are used to solve the design problem in terms of assigning data flow to channels and task to nodes (5). UML diagrams are parsed to generate a set of equations which can be solved by SAT solvers and the results are back-annotated into UML diagram.

Furthermore, it could be interesting to test the analytical solutions by using simulation. For this purpose, the network design is extracted from the UML model (6) and represented by the intermediate representation and then an executable

network scenario is generated from it (7); nodes and channels are modeled by using the SystemC Network Simulation Library (SCNSL) [6] which reproduces the behavior of packet-based networks over SystemC and generates the corresponding statistics (e.g., packet loss rate and delay). The statistics are used to choose between different analytical solutions (8).

Moreover, HIF-based network description is manipulated according to mathematical-based rules to obtain several alternatives which are equivalent from the network perspective (9). Each alternative is simulated again by using the network simulator (SCNSL). Simulation results are used to choose between different alternatives. Another feature added to this framework is a methodology to generate an abstracted description of RTL IPs into HIF (label A) and then translate it to UML (B). By using such methodology, UML models can be automatically created starting from existing RTL IPs [4].

III. ACHIEVEMENTS, CURRENT AND FUTURE WORK

The proposed methodology has been accepted and published in [1]. Methodology to generate SystemC/TLM and RTL code with checkers from UML/MARTE diagrams has been published in [2], [3]. Methodology to abstract RTL models and representing it in a higher level of abstraction has been accepted in [4].

Based on the previous work, tools have been developed for automatic code generation from UML diagrams such as UML2HIF front-end tool and to manipulate HIF description. Journal paper has been submitted and UML profile for network modeling has been developed. The proposed methodology has been developed in strict relationship with two European projects. *COMPLEX project*: RF module has been implemented and tested for a health care application [5]. *TouchMore project*: methodology to abstract RTL IPs into UML/SysML has been set and a tool has been developed [4].

Current work consists in developing custom profiles to give a precise semantics to UML diagrams and a set of tools for automatic code generation from UML diagrams for simulation. Future work will consist in automatic exploration of the network design space to automatize the synthesis process by using SAT solver along with HIF and the computational model.

REFERENCES

- [1] E. Ebeid, F. Fummi, D. Quaglia, and F. Stefanni. Refinement of UML/MARTE models for the design of networked embedded systems. In *ACM/IEEE DATE'12*, pages 1072–1077, 2012.
- [2] E. Ebeid, D. Quaglia, and F. Fummi. Generation of SystemC/TLM code from UML/MARTE sequence diagrams for verification. In *IEEE DDECS'12*, pages 187–190, 2012.
- [3] E. Ebeid, D. Quaglia, and F. Fummi. Generation of VHDL code from UML/MARTE sequence diagrams for verification. In *Euromicro DSD'12*, 2012.
- [4] E. Ebeid, N. Bombieri, M. Lora, and F. Fummi. On the reuse of RTL IPs for SysML model generation. In *IEEE MTV'12*, 2012.
- [5] M. Lazarescu, P. Sayyah, D. Quaglia, E. Ebeid, S. Bocchio, and A. Rosti. Network-aware Design-Space Exploration of a Power-Efficient Embedded Application. In *ACM/IEEE CODES*, 2012.
- [6] F. Fummi, D. Quaglia, and F. Stefanni. A SystemC-based framework for modeling and simulation of networked embedded systems. In *ECSI FDL*, pages 49–54, 2008.
- [7] F. Fummi, D. Quaglia, and F. Stefanni. Modeling of communication infrastructure for design-space exploration. In *ECSI FDL*, 2010.
- [8] F. Leonardi, A. Pinto, and L. P. Carloni. Synthesis of distributed execution platforms for cyber-physical systems with applications to high-performance buildings. In *ACM/IEEE ICCPS*, pages 215–224, 2011.