Microprocessors and Microsystems 39 (2015) 843–853

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

Simulation alternatives for the verification of networked cyber-physical systems

Michele Lora[a,*], Riccardo Muradore[a], Davide Quaglia[a,b], Franco Fummi[a,b]

[a] Department of Computer Science, University of Verona, Italy
[b] EDALab s.r.l. - Verona, Italy

ARTICLE INFO

Article history:
Received 9 February 2015
Revised 5 August 2015
Accepted 8 September 2015
Available online 19 October 2015

Keywords:
SystemC-AMS
Network simulation
Co-simulation

ABSTRACT

Several embedded system applications are used to control physical processes thus leading to the concept of Cyber-Physical System. Sensing, computation and actuation are combined by involving a set of highly heterogeneous components, *i.e.*, digital and analog hardware, software, energy sources, and external physical systems (*e.g.*, a room to be heated, a mechanical component). Moreover, we see the growing use of networks to connect several sub-systems in a more efficient way; this contributes to introduce a further level of heterogeneity due to the presence of messages exchanged over communication channels. All these aspects should be taken into account in the design process to find highly optimized solutions; in particular, simulation is a key technique for their verification. However, the heterogeneity of components, together with the presence of the network, forces to adopt complex and slow co-simulation techniques to carry on the simulation of the entire system.

This work aims at proposing SystemC as unified framework to model and simulate Networked Cyber-Physical Systems. Concerning the modeling of continuous-time components and a specific class of discrete-time components, the different Models of Computation provided by the Analog/Mixed-Signal (AMS) extension of SystemC are used. Regarding the network, SystemC and the SystemC Network Simulation Library are used to model communications at different abstraction levels. The approach is validated by considering a networked control system. First, the accuracy and speed of SystemC simulation is compared to Matlab/Simulink simulation. Second, different simulation alternatives are compared to co-simulation. Finally, we show that the simulation speed-up achieved by all-SystemC modeling allows to explore several design alternatives and the accurate modeling of the communication network can improve the design of the controller.

1. Introduction

Nowadays, embedded systems are widely used to manage physical processes thus leading to the concept of Cyber-Physical Systems (CPS) [17]. Usually, the physical elements to be controlled are inserted into a loop together with sensing, computation and actuation processes. On one hand, this gives the chance to embed a form of "intelligence" into the physical world, thus allowing to fully control it. On the other hand, the design of these complex systems requires to deal with different components belonging to a heterogeneous set of domains. Moreover, it can be often convenient to place the computational resources remotely with respect to the physical processes to control thus leading to the introduction of network connections that also contribute on increasing the level of heterogeneity. Let us denote such systems as Networked Cyber-Physical Systems. Fig. 1 depicts an example of such distributed systems. Usually they feature a hierarchical structure in which fine-grained control tasks are configured and supervised by high-level applications. These two levels are connected by a traditional packet-based network like Ethernet. Control tasks consist of the usual feedback loop in which the controller sends commands to the actuators on the system to be controlled (denoted as "plant"); the controller computes command values according to measurements about the plant made by sensors. In recent years, packet-based networks have been introduced to close the loop between controller and plant, *e.g.*, when the controller should be positioned far from the plant for security reasons or when multiple control loops share the same communication infrastructure. In this case, we speak about Networked Control Systems (NCS).

In Fig. 1, two NCSs are integrated with normal computer equipments over Ethernet network (*e.g.*, a Supervisor, a data Logger, a Database server, and a Firewall for Internet connection). In each NCS, measurements and commands are transmitted through a

---

* Corresponding author.
E-mail addresses: michele.lora@univr.it (M. Lora), riccardo.muradore@univr.it (R. Muradore), davide.quaglia@univr.it, davide.quaglia@edalab.it (D. Quaglia), franco.fummi@univr.it, franco.fummi@edalab.it (F. Fummi).

http://dx.doi.org/10.1016/j.micpro.2015.09.001
0141-9331/

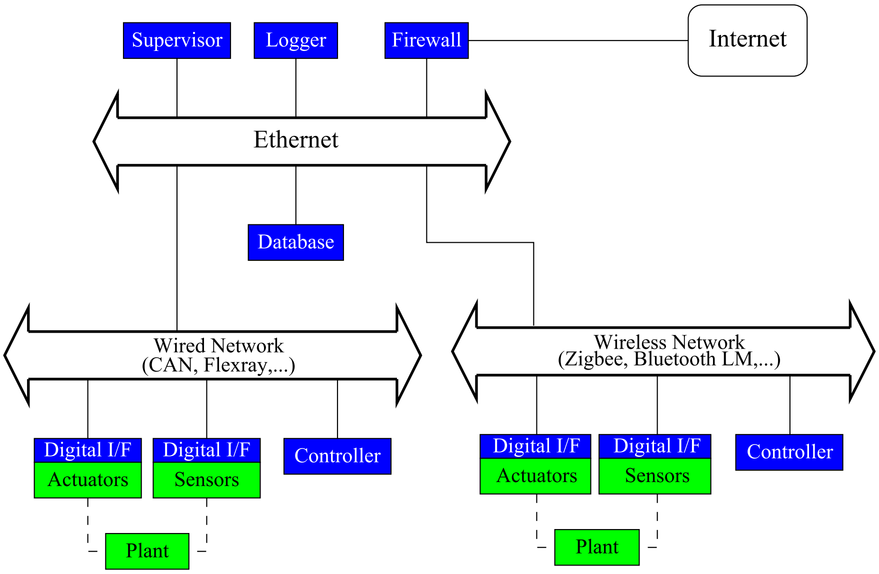

Fig. 1. Example of Networked Cyber-Physical System.

packet-based network which can be either wired (*e.g.*, CAN, Flexray) or wireless (*e.g.*, ZigBee, Bluetooth).

The design of such systems requires a wide set of different domain-specific tools and methodologies handling their heterogeneity. To further push optimization of design solutions, a holistic approach is needed in particular to verify the whole system. Simulation is the best approach in case of complex systems. The difficulty here is that every design domain usually addresses the problem of the simulation by employing ad-hoc tools and the corresponding Models of Computation (MoC). For instance, recalling Fig. 1, green boxes represent physical components which can be described by using *continuous-time models* while blue boxes represent digital components which can be described by using *discrete-time models*. Sensors and actuators have a digital interface for the connection to the network.

The continuous-time domain is related to non-electrical components and analog hardware which can be modeled by systems of differential equations, while the discrete-time domain is related to digital hardware and software which are modeled by state machines or difference equations.

Concerning the network (white components in Fig. 1), the problem is further enlarged since *network* communications can be modeled at multiple levels, considering physical signals, bits, or packets. While the behavior of physical level can be described by continuous-time models, at upper levels discrete-time models are more suitable.

It is current state of the practice to simulate each domain (continuous-time, discrete-time and network) by employing ad-hoc simulators and combining them together by using interfaces for synchronization. Such interfaces present two major drawbacks. The first one is the overhead introduced for the communication between different tools. The second one is the possible introduction of additional design error sources due to the difficulties to formally define the behavior at the interface between different domains.

This work aims at overcoming this limitations by proposing a framework that allows to move from co-simulation to *holistic simulation* for Networked CPS. The main contributions of the work in this direction are:

- A set of methodology to use SystemC [2], its extension for analog/mixed-signals (AMS) components [1], and the SystemC Network Simulation Library (SCNSL) [10] as a single framework to represent discrete-time, continuous-time and network components in a unique model.
- A packet-based representation of the network, with respect to traditional low-level channel modeling (e.g., provided by Matlab Telecommunication Toolbox and [40]). Some alternatives are proposed for packet-based simulation, *i.e.* SCNSL, SystemC-AMS Timed Data Flow (TDF) MoC, and SystemC-TLM. We will show that such simulation alternatives are characterized by a different trade-off between accuracy and simulation speed.
- A further analysis of the proposed alternatives shows that simulation speed also depends on the mix of the techniques used to model the different parts of the systems. The analysis will focus on the advantages and drawbacks of using different available MoCs to model the functionality of the system.

The paper is organized as follows. Sections 2 and 3 formulate the problem and give an overview of previous results concerning the simulation of Networked CPS. Section 4 introduces the SystemC extensions used in this work (*i.e.*, SystemC-AMS and SCNSL). Section 5 describes the proposed methodologies. Section 6 presents their application to a case study. Finally, some concluding remarks are drawn in Section 7.

2. Problem formulation

Continuous-time models represent system evolution by using a set of differential equations or, in case of linear time-invariant systems, Laplace transfer functions. The behavior of the system is the solution of such equations and consists of a set of continuous temporal functions describing the evolution of system output as depicted in Fig. 2. Even if every system has intrinsically a continuous-time behavior, for some systems we are interested in knowing their state for specific time values. In this case, discrete-time models are a more efficient way to describe their behavior. For example, in digital hardware and software, the state consists of the content of internal memory and variables, respectively. Their value changes when specific events happen and, therefore, in specific time instants. It is worth noting that, in general, these instants are not equally spaced. This behavior can be represented by using *event-driven models* such as state charts and Petri Nets depicted in Fig. 3.

For a particular sub-set of discrete-time systems, state value changes after constant time intervals. For example, digital signal processing tasks and digital controllers process data samples at a