Simulation Modelling Practice and Theory 43 (2014) 16–33

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

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

A co-simulation framework for design of time-triggered automotive cyber physical systems

Zhenkai Zhang *, Emeka Eyisi, Xenofon Koutsoukos, Joseph Porter, Gabor Karsai, Janos Sztipanovits

Institute for Software Integrated Systems (ISIS), EECS Vanderbilt University, Nashville, TN, USA

A R T I C L E I N F O

Article history:
Received 24 August 2013
Received in revised form 18 December 2013
Accepted 6 January 2014
Available online 29 January 2014

Keywords:
Co-simulation
Virtual prototyping
Model-based design
CPS
Automotive control system
SystemC

A B S T R A C T

Designing cyber-physical systems (CPS) is challenging due to the tight interactions between software, network/platform, and physical components. Automotive control system is a typical CPS example and often designed based on a time-triggered paradigm. In this paper, a co-simulation framework that considers interacting CPS components for assisting time-triggered automotive CPS design is proposed. Virtual prototyping of automotive vehicles is the core of this framework, which uses SystemC to model the cyber components and integrates CarSim to model the vehicle dynamics. A network/platform model in SystemC forms the backbone of the virtual prototyping. The network/platform model consists of processing elements abstracted by real-time operating systems, communication systems, sensors, and actuators. The framework is also integrated with a model-based design tool to enable rapid prototyping. The framework is validated by comparing simulation results with the results from a hardware-in-the-loop automotive simulator. The framework is also used for design space exploration (DSE).

1. Introduction

Cyber-physical systems (CPS) are complex systems that are characterized by the tight interactions between the physical dynamics, computational platforms, communication networks, and control software. When designing CPS, a practical approach is to consider the physical layer, the network/platform layer, and the software layer, as shown in Fig. 1 [1]. The physical layer represents physical components and their interactions, whose behavior is governed by physical laws and is typically described in continuous time using ordinary differential equations. The network/platform layer represents the hardware and includes the network architecture and computation platform that interact with the physical components through sensors and actuators.

As a classical CPS domain, automotive systems have been gaining a lot of attention. As automotive system functionalities are increasingly implemented by electronic instead of hydraulic or mechanical systems, up to 70 electronic control units (ECUs) exchanging more than 2500 signals over up to 5 different communication systems can be found in a modern vehicle [2]. The complex cyber-physical interactions make the composability and predictability of these safety–critical systems very challenging. Furthermore, the economy factors, such as persistent effort for low production costs and tight time-to-market, further complicate the design of such systems.

* Corresponding author. Tel.: +1 615 322 8477.
E-mail addresses: zhenkai.zhang@vanderbilt.edu (Z. Zhang), emeka.eyisi@vanderbilt.edu (E. Eyisi), xenofon.koutsoukos@vanderbilt.edu (X. Koutsoukos), joe.porter@vanderbilt.edu (J. Porter), gabor.karsai@vanderbilt.edu (G. Karsai), janos.sztipanovits@vanderbilt.edu (J. Sztipanovits).

1569-190X/$ - see front matter
http://dx.doi.org/10.1016/j.simpat.2014.01.001

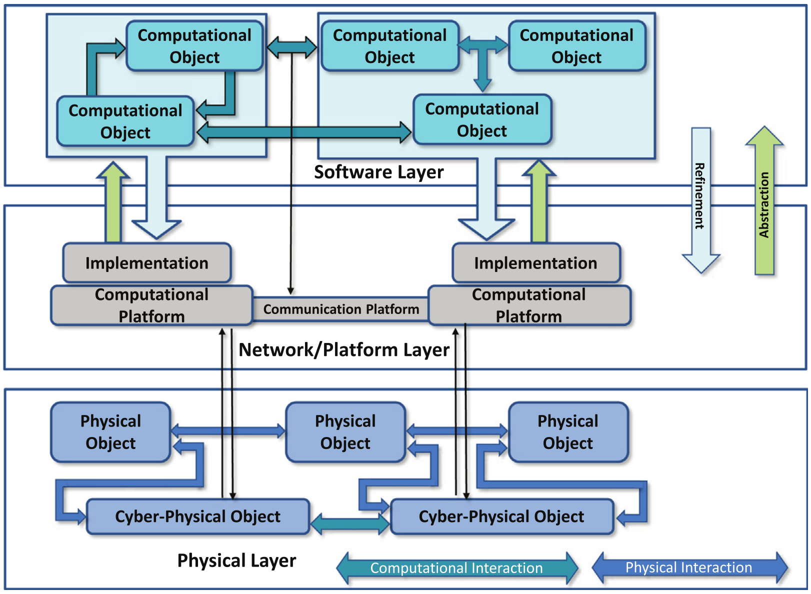

Fig. 1. A simplified view of designing CPS: Three CPS design layers [1].

Time-triggered architecture (TTA) has been proposed and widely used to address the complexity and composability difficulties posed by automotive control systems by precisely defining the interfaces between components both in the time and value domain in order to provide predictability [3]. In addition, there have been on-going efforts towards the standardization of in-vehicle communication systems based on time-triggered (TT) paradigms (e.g. FlexRay and TTEthernet) with the overall goal of ensuring highly reliable, deterministic, and fault-tolerant system performance [4,5].

The layered CPS design approach can be easily applied to TT automotive design. We often start designing the automotive control system using a high level modeling language such as MATLAB/Simulink [6]. The model serves as an executable specification and the equivalent source code, usually in C, can be generated automatically from the model. At later design stages, the generated source code is deployed on an automotive vehicle platform to perform the required functionality. It may not be possible to achieve the required control performance if system elements are designed separately and integrated in the end. Interactions between the layers are very tight, so late integration is very likely to result in large design gaps that will be costly to resolve. Moreover, different design options (e.g. processors, communication systems, and software deployments) may need to be explored in order to find trade-offs between performance and economy factors. In order to reduce the effort and cost as well as shorten time-to-market, it is important to enable design space exploration (DSE) and get realistic control performance feedback at early design stages. However, the vehicle platform prototype is usually not available at early design stages and even if it is available, testing at the very beginning presents safety and economical challenges.

A cross-layer co-simulation framework that takes into account physical dynamics, control software, computational platforms, and communication networks becomes crucial in the design of automotive CPS. The requirements for such a framework include: (1) it should contain models from all design layers that can be integrated together; (2) the models should be at appropriate levels of abstraction, so that the simulation is efficient but accurate enough; (3) the scalability of the framework should allow simulation of large distributed automotive CPS; (4) it should allow model-based rapid prototyping to improve the usability.

Co-simulation can be achieved by virtual prototyping. Virtual prototyping can take advantage of different modeling languages/tools and integrate them together to evaluate the whole CPS. Modeling cyber components in SystemC has begun to be dominant in the Electronic System-Level (ESL) design field. SystemC has become a *de facto* system-level design language for hardware/software (HW/SW) co-design and an IEEE standard [7]. SystemC allows modeling at different levels of abstraction. By adding appropriate timing annotations, a SystemC model can reveal timing behavior of the corresponding HW/SW. SystemC also has a standardized library for realization of transaction level modeling (TLM) concepts. TLM focuses on what data is being transferred rather than how it is being transmitted, so a TLM model abstracts away certain communication details to speed up simulation while keeping sufficient accuracy.

The main contributions of the paper include: (1) A co-simulation framework for design of time-triggered automotive CPS that centers on a detailed network/platform layer model in SystemC is proposed. The network/platform layer model, including processing elements (PEs) which are abstracted by real-time operating system (RTOS) models, TTEthernet communication systems, sensors, and actuators, enables TT computation and communication; (2) Rapid prototyping is realized by model transformations from a designed MATLAB/Simulink model to a front-end design environment model to the final virtual prototype. It enables fast generation of executable simulation models with different configurations, including hardware