Generic Framework for Design, Modeling and Simulation of Cyber Physical Systems

Ji Eun Kim[†‡], Daniel Mosse[†]
Computer Science Department, University of Pittsburgh[†]
Research and Technology Center, Robert Bosch LLC[‡]
{kim, mosse} @cs.pitt.edu

Abstract

Wireless technologies have contributed to extensive development of many kinds of applications that enable cost-effective and intelligent monitoring and control of physical environments, objects, activities such as smart buildings, home or industry automation, and remote patient care. These kinds of applications often require support of heterogeneous physical environments and large-scale systems mixing several different types of monitoring and control. In addition to heterogeneity and scalability support, mobility support is crucial in the development of these software systems.

Although much research has addressed design, modeling and simulation of such systems, most solutions usually do not cover all requirements from different domains. Rather, these solutions restrict the development environment and reduce flexibility of design by enforcing the use of a specific software platform and tightly-coupled tools.

This paper presents problems and challenges of Cyber Physical Systems (CPS) like the aforementioned applications, and discusses features and benefits of a generic framework to enable design, modeling and simulation of large-scale, heterogeneous CPS systems in an integrated manner.

1. Introduction

Wireless technologies have been extensively used for many years in a myriad of applications, often used to monitor and control various physical environments, objects and activities. Cyber Physical Systems (CPS) is a new paradigm extending the use of such technology, combining the control of embedded systems, wireless networking and real-time systems. For example, smart building, home or industry automation and remote patient care are emerging markets that need such applications.

The software development environment should be quite different from conventional embedded software development because of many reasons, including that a CPS usually consists of many sensors and actuators communicating with each other, and CPS software development requires the modeling of *external resources* such as light source, temperature, and signals. Furthermore, most CPSs are *resource constrained* in terms of memory, processing power, and battery life. Because of that, the development environment should provide software architectural and design solutions to minimize resources, and an accurate simulation environment to support *battery life* and *timing related communication*. In addition, the nature of *heterogeneity* and *scalability* introduce a big challenge to deploy, test and maintain CPS. Thus, accurate modeling and simulation environment can reduce the effort and cost of deployment, testing and maintenance. Lastly, the domains using CPS often require *mobile nodes* that must interact in a scalable manner with the various nodes in the system.

2. Problems

TinyOS [1] and Zigbee-based wireless sensor networks applications have been extensively researched in academia and industry. Although both technologies expedite the development of wireless sensor networks applications, many applications used in the industry still require the use of their proprietary solutions including software platforms and wireless protocols. Reasons of this tendency can include: 1) legacy software (e.g., wired communication software) integration with new sensor applications, 2) cost issues that require optimizing resources (e.g., memory and communications), and 3) modifiability requirements to enable adding and adapting various requirements easily.

This tendency limits the use of approaches and tools developed for specific software platforms and wireless protocols. For example, Viptos [2] has been developed to provide a graphical environment for modeling and simulating wireless sensor applications mainly using TinyOS. It integrated TOSSIM [3] engines, which are also coupled to TinyOS, to support the simulation of heterogeneous and large scale sensor applications. Although the features provided in such a tool are attractive to users, it requires tremendous efforts to adapt it for different users' development environments.

In order to make available solutions that are more applicable to various domains, a generic framework is greatly needed to support easy adaptations to proprietary solutions, including different types of hardware, different physical modeling environments, and different platforms, and emulation/simulation of various use-cases.

3. Features needed in CPS frameworks

The following features are essential in a generic framework for CPS design, modeling, and simulation.

- ***Heterogeneous (in terms of various types of sensors/actuators) applications support***: CPS usually consists of non-homogeneous applications. Thus, it should be able to simulate heterogeneous application logics simultaneously.
- ***Various physical modeling environments:*** the physical modeling environment should support mathematical expressions and incorporate domain specific physical modeling descriptions (e.g. floor plan of buildings) by extracting relevant information from them.
- ***Scalability support:*** support for the development and simulation ranging from small scale (tens) to large scale (thousands of) sensors and actuators.
- ***Mobility support:*** support for modeling systems using relevant properties (e.g., communication, signal strength).
- ***Integration of existing simulation tools:*** easy-to-use support to link to existing simulation tools is required.
- ***Integration of proprietary solutions and open standards support:*** proprietary solutions and open standards including protocols, infrastructures and existing software should be able to be easily incorporated into a generic framework.
- ***Software reuse:*** a generic framework should support software reuse either by exploiting code generation techniques (which can also use proprietary infrastructure), linking libraries or using configurable components.
- ***Usability:*** Graphical representation of modeling and simulation environment can enable easy development of new applications. Domain-specific 3D modeling environments can also be supported depending upon requirements.

4. Challenges

Generic solutions are often thought not to be useful for most applications, because generality is typically associated with "not-specific enough" to be ready for use. Thus, it is important at the beginning of research to investigate the characteristics of CPS to scope the target domains, elicit exhaustive use-cases, and develop the software architecture and solutions for a generic framework.

It is a challenge how we can effectively use and integrate existing solutions. Simulation engines, modeling techniques, software engineering methodologies, proprietary infrastructure and open standards were not intended to be used for a consistent development environment. Thus, integration and development efforts should be justified through the evaluation of benefits of a generic framework using elicited use-cases upfront.

5. Benefits

Our concept of a generic framework can provide seamless development by supporting different levels of modeling, design and simulation in an integrated manner. This generic framework can be a possible direction to minimize effort to test heterogeneous and large scale sensor systems by exploiting accurate simulation. It can also reduce development efforts by enabling software reuse of open source or in-house software infrastructures and communication protocols, and accurate simulations.

6. Conclusion

This paper discusses problems and challenges in the software development of CPS systems. We propose to develop a generic framework to enable design, modeling and simulation of CPS by eliciting important features and anticipated benefits. This generic framework can support effective use of existing software, rapid introduction and integration of new standards, protocols and software platforms through the support of a consistent development environment.

7. References

[1] TinyOS, http://www.tinyos.net/

[2] Elaine Cheong, Edward A. Lee and Yang Zhao. "Joint Modeling and Design of Wireless Networks and Sensor Node Software," *UCB/EECS-2006-150, November 17, 2006.*

[3] Levis et al, "TOSSIM: Accurate and Scalabe Simulation of Engire TinyOS Aplication", *Embedded Networked Sensor Systems (SenSys 2003).*