Providing End-to-End Guarantees in Cyber-Physical Systems

Harini Ramaprasad, Southern Illinois University Carbondale

October 21, 2008

1 Introduction

Cyber-physical systems are integrations of computation, communication and control with physical processes/entities. They may be viewed as a networked system of embedded systems that must work together in a safe, efficient, reliable, predictable and timely manner to monitor and control physical entities.

In recent times, embedded systems have become ubiquitous and a significant amount of research has been and is being conducted to make a single embedded system safe, efficient and reliable. However, cyber-physical systems take this one step further and integrate multiple, potentially diverse, embedded systems with physical entities, thus exacerbating some of the challenges faced by real-time/embedded systems designers and introducing several new ones. In addition to providing guarantees of temporal correctness within one embedded system, end-to-end guarantees must now span across embedded systems via a communication network.

This paper briefly discusses some of the challenges involved in the design, development and deployment of cyber-physical systems.

2 Challenges in Cyber-Physical Systems

The three basic functional components of a cyber-physical system include computation, communication and control. Each of these components operates along multiple dimensions that must be abstracted from the other dimensions, thus enabling the modular modeling of the complete, multi-domain system. One of the dimensions is time. A primary requirement of a cyber-physical system is that it has to operate in a timely manner. In other words, the system has to adhere to real-time constraints. Fundamentally, this translates to real-time requirements in each of the components at the hardware and software levels.

2.1 Real-Time Computation

Providing timing guarantees for computation within an embedded real-time system has been the focus of much research over the last several years. Traditionally, much of the “normal” behavior

in embedded real-time systems has been modeled as a set of periodic tasks that have specific activation frequencies. External events that occur at unknown times, but whose timing characteristics become known upon release, are modeled as sporadic jobs.

Since sporadic jobs by definition arrive at arbitrary times, their incorporation into the system is not guaranteed. Instead, an "acceptance test" is conducted in order to determine whether a sporadic job may be accepted without causing the existing jobs to violate their temporal constraints. If a sporadic job fails the accpetance test, depending on the importance of the job, the system might shift into a "recovery" mode.

While this behavior might be acceptable for closed embedded real-time control systems, it could be a much more severe problem in the context of cyber-physical systems. By definition, a cyber-physical system interacts closely with physical entities and the physical environment. The physical environment is dynamic by nature and, hence, it will become much harder to model even the "normal" behavior of cyber-physical systems in a time-driven manner using periodic tasks. For instance, it might not be feasible to jeopardize the normal operation of the system and switch to a recovery mode in order to handle sporadic jobs that fail the acceptance test. **This calls for rethinking of approaches towards event-driven modeling of real-time tasks**. A starting point for this would be to develop techniques to **incorporate sporadic jobs by design rather than by chance**.

One way of improving the likelihood of acceptance of sporadic jobs is to use sporadic servers [3]. The basic idea here is to model the sporadic server as a periodic task for the purposes of design and analysis of the system. However, this introduces several **challenges in timing analysis** of real-time systems, which is imperative to provide temporal guarantees. Most existing methodologies and tools for timing analysis in the context of multi-task environments ([1, 4, 2]) assume that all tasks are periodic and that the timing and caching behavior of all these tasks are known *a-priori*. When a sporadic server is modeled as a periodic task, although the execution time of the server is constrained and predictable, the behavior of the actual sporadic job executing in that time with respect to architectural features such as the pipeline and the cache are not known. This necessitates the **development of offline-assisted techniques to bound the behavior of sporadic jobs at run-time** with respect to the other tasks in the system, which poses a significant research challenge.

2.2 Predictable Communication

Networking between various potentially diverse systems is at the center of a cyber-physical system. There are potentially three levels of networking involved a cyber-physical system. 1) **Network on Chip (NoC)**: These days, many multicore embedded systems are being deployed. In order to provide predictable and scalable communication between the various cores and other modules such as the memory module, caches, etc., network on chip is a viable option. 2) **Communication within a closed system**: A single embedded system, such as an automobile or an aircraft, consists of several electronic control units (ECUs), each performing certain tasks. These ECUs communicate with each other and with peripherals such as sensors and actuators through a communication network. 3) **Communication across multiple systems**: Many embedded systems interact with each other and with the physical environment to form the final cyber-physical system. For example, a vehicle-to-vehicle system.

While the basic goal of all three levels is secure and reliable communication, they each present very different challenges to system designers in terms of the type of communication used (wired vs. wireless), in terms of the security required (communication across multiple systems requires more stringent security protocols) and in terms of timing predictability (each level operates at significantly different granularities of time). The first two levels of communication have been and are being reasonably well researched in the context of real-time/embedded systems. However, the third level is largely unexplored, requiring concerted research efforts. Providing **end-to-end timing guarantees** at this level remains a challenging problem.

In much the same way as general networking techniques have been adapted to micro-level communication on a single chip, leading to NoCs, some of the **techniques used in wireless sensor networks may be leveraged and adapted to a macro-level** in the context of communication across multiple embedded systems. The fundamental distinction between the communication in wireless sensor networks and cyber-physical systems is the **lack of homogeneity** among the communicating nodes in the case of a cyber-physical system. In this context, it is imperative to **establish a protocol of communication** between diverse systems in various domains of science and engineering.

3 Author Biography

Harini Ramaprasad, Assistant Professor of Electrical and Computer Engineering, Southern Illinois University Carbondale. E-mail: harinir@siu.edu Phone: 919-522-6293.

Harini Ramaprasad has been an Assistant Professor at Southern Illinois University Carbondale since August 2008. She received her Ph.D. in Computer Science from North Carolina State University in July 2008. Her research interests include real-time/embedded systems, cyber-physical systems, operating systems, compilers and computer architecture.

References

[1] C.-G. Lee, K. Lee, J. Hahn, Y.-M. Seo, S. L. Min, R. Ha, S. Hong, C. Y. Park, M. Lee, and C. S. Kim. Bounding cache-related preemption delay for real-time systems. *IEEE Transactions on Software Engineering*, 27(9):805–826, Nov. 2001.

[2] H. Ramaprasad and F. Mueller. Tightening the bounds on feasible preemptions. *ACM Transactions on Embedded Computing Systems*, page (accepted), Mar. 2008.

[3] B. Sprunt and L. Sha. Scheduling sporadic and aperiodic events in a hard real-time system. Technical Report CMU/SEI-89-TR-11 ADA211344, Software Engineering Institute (Carnegie Mellon University), 1989.

[4] J. Staschulat and R. Ernst. Multiple process execution in cache related preemption delay analysis. 2004.