Editorial: Reflections on the History of Cyber-Physical versus Embedded Systems

I am often asked by students to differentiate between "embedded systems" and "cyber-physical systems." I often answer in jest: "Aren't there two distinct ACM Transactions on each of these topics? We also have embedded systems week (ESWEEK), a federated conference on embedded systems, and cyber-physical systems week (CPSWEEK), another federated conference on cyber-physical systems."

Jokes aside, I think, in the beginning, there was some confusion—after all, the U.S. National Science Foundation coined the term "cyber-physical systems (CPS)" in the early 2000s—and the funding program had grown out of their "embedded and hybrid systems" program. Over the past two decades, CPS has grown as a discipline of its own, but the two disciplines are intricately intertwined.

In the 1990s, we used to define embedded systems as any computing system that was embedded inside an equipment that was not a traditional computer. Examples would be digital camera, CD player, a drive-by-wire automotive, or a fly-by-wire airplane. A lot of industrial automation components, such as sensors, actuators, and equipment like digital relays in power grid, would be considered embedded systems. They had microprocessors, microcontrollers, signal-processing chips, firmware, and rudimentary software running on them to provide functionality. Over time, almost every piece of equipment started to possess some computing facilities—and there was a widely read report commissioned by the National Research Council in the U.S. in 2001 that outlined the various ways our daily lives have been run and influenced by the embedded devices. This is long before the advent of smart phones, smart television, or even macroscale concepts, such as smart city, smart transportation, smart grid, and so on. The concept of hybrid systems that had digital controllers interacting through sensors/actuators with the physical dynamics of a physical process was present in the early 1990s, and various formal models, such as Hybrid automata and their verification methodologies, were heavily researched. However, such models did not scale very well at that time, and state-space abstraction techniques with model checking of hybrid systems became a fashionable research topic for a while.

I recall, in 2001, we met at Lake Tahoe for the first Conference on Embedded Software (EMSOFT), which was based on several invited talks—many of which focused on design platforms, languages, and verification of embedded software—although there was the demonstration of smart motes in the same conference. This is when embedded computing started coming out of individual devices, and a need for modeling the interaction between embedded devices on a network and their interaction with the physical dynamics and human intervention started to take shape.

I believe that around 2003 the National Science Foundation first released the call for proposals for cyber-physical systems, and there was a lot of confusion as to how it differed from the embedded and hybrid systems. In hindsight, it seems that the difference was in the scale, scope, and vision.

It was clear that as technologies, power grids, and manufacturing were becoming computer controlled and sensor-based, there was also a lot more scope of optimal control, if control was at the global scope with global visibility, and it required a much larger-scale modeling and, accordingly, the methodology to process and verify such models. One technology that was being played out at

1539-9087/2019/06-ART19e
https://doi.org/10.1145/3325115

that time was the increasing use of middleware and standardization of frameworks such as CORBA. The seamless integration of heterogenous software components—and thereby increasing interaction between hardware components—allowed the scope to enlarge to the macroscoping scale. The interaction with physical dynamics and measurements thereof would make the modeling more challenging.

Before this, most of the industrial-scale systems had a siloed approach to design—the hardware was designed separately, the physical dynamics was understood separately, and the control was designed based on the model of the physical dynamics. Since differential equations compose naturally, composition of models was not an issue. Once the control design was completed and algorithms written, only then would implementation in microcontroller/processors occur. These silos created a bottleneck to achieving a global model, and hence global optimization scopes were reduced. Optima of individual components do not compose to give global optima, nor do models of finite-state systems and differential equation systems compose naturally. We needed to evolve the science of cyber-physical systems.

However, at the individual component level, we still need to focus on embedded computing, hardware design, firmware issues, software stack, and networking interfaces. We need to improve security of individual devices. In our lab, in some of the components of a power system test bed—which include PLC, protocol converter switches, power meters, and protocol implementation of device level interfaces—all have been found to have a multitude of security vulnerabilities. Most of these are at the software and firmware level. However, researchers working on side-channel analysis find problems with respect to hardware components as well. The power consumption at the device level is still be a big issue. As transistors are scaling down, design automation methodology and tools—manufacturing variations are affecting yields—are equally important. Therefore, embedded-system research is still important, as is the research in cyber-physical systems. The two can complement each other rather than compete as disciplines to attract the attention of researchers.

One might wonder why I am writing about this. In a recent review of ACM journals, it came to our attention that the number of submissions to ACM TECS has gone down drastically since around 2016. One reason, of course, is that earlier we published many special issues (more than one special section per issue), mostly based on papers chosen from conferences. In the past two years, we have published much fewer special issues—ever since ACM TECS started to be the publication venue for ESWEEK conferences under the "journal-first publication" model. However, the other reason is that a new ACM Transactions, specifically meant for cyber-physical systems, has started—and a lot of the articles on CPS that used to have only one venue for publication have found a better home now. This thought led me to reflect on the relationship between these two disciplines and how they complement each other—and quite naturally, it made me reflect on the early days of CPS research. Of course, another researcher may remember the history in a very different way.

Coming back to ACM TECS 18:3, we are publishing a special issue on Cryptographic Engineering for the Internet of Things. I must thank Prof. Lejla Batina, Prof. Sherman S. M. Chow, Prof. Gerhard Hancke, and Prof Zhe Liu for their dedication and hard work in making this special issue possible. The intent, and the selection of articles for the special issue, is detailed in their guest editorial.

Sandeep K. Shukla
Indian Institute of Technology Kanpur
Editor-in-Chief