CHALLENGES AND OPPORTUNITIES FOR SYSTEM THEORY IN EMBEDDED CONTROLLER DESIGN

Alberto Sangiovanni Vincentelli *

* *The Edgar L. and Harold H. Buttner Chair, Department of Electrical Engineering and Computer Science, University of California at Berkeley*

Abstract: Embedded controllers are essential in today electronic systems to assure that the behaviour of complex systems as cars, airplanes, trains, building security management systems, is compliant to strict safety constraints. I will review the evolution of embedded systems and the challenges that must be faced in their design. I will also present methodologies aimed at simplifying and speeding the design process. The role of hybrid systems in the development of embedded controllers will be outlined. Future applications such as wireless sensor networks in an industrial plant will also be presented. *Copyright © 2006 IFAC.*

Keywords: Embedded Systems, Systems Design, Systems Methodology, Control Applications, Distributed Control.

EXTENDED ABSTRACT

The ability of integrating an exponentially increasing number of transistors within a chip, the ever-expanding use of electronic embedded systems to control increasingly many aspects of the ”real world”, and the trend to interconnect more and more such systems (often from different manufacturers) into a global network, are creating a nightmarish scenario for embedded system designers. Complexity and scope are exploding into the three inter-related but independently growing directions mentioned above, while teams are even shrinking in size to further reduce costs. In this scenario the three challenges that are taking center stage are:

- *Heterogeneity and Complexity of the Hardware Platform.* The trends mentioned above result in exponential complexity growth of the features that can be implemented in hardware. The integration capabilities make it possible to build real complex system on a chip including analog and RF components. The decision of what to place on a chip is no longer dictated by the amount of circuitry that can be placed on the chip but by reliability, yield and ultimately cost (it is well known that analog and RF components force to use more conservative manufacturing lines with more processing steps than pure digital ICs). Even if manufacturing concerns suggest to implement hardware in separate chips, the resulting package may still be very small given the advances in packaging technology yielding the concept of System-in-Package (SiP). Pure digital chips are also featuring an increasing number of components. Design time, cost and manufacturing unpredictability for deep submicron technology make the use of custom hardware implementations appealing only for products that are addressing a very large market and for experienced and financially rich companies. Even for these companies, the present design methodologies are not yielding the necessary productivity forcing them to increase beyond reason the size of design and verification teams. These IC companies (for example Intel, AMD and TI) are looking increasingly to system design methods to allow them to assemble large chips out of pre-designed components and to reduce validation costs. In this context, the adoption of design models above RTL and of communication mechanism among components with guaranteed properties and standard interfaces is only a matter of time.

- *Embedded Software Complexity.* Given the cost and risks associated to developing hardware solutions, an increasing number of companies is selecting hardware platforms that can be customized by reconfiguration and/or by software programmability. In particular, software is taking the lion's share of the implementation budgets and cost. In cell phones, more than 1 Million lines of code is standard today, while in automobiles the estimated number of lines by 2010 is 100 Millions. The number of lines of source code of embedded software required for defense avionics systems is also growing exponentially. However, as this happens, the complexity explosion of the software component causes serious concerns for the final quality of the products and the productivity of the engineering forces. In transportation, the productivity of embedded software writers using the traditional methods of software development ranges in the few tens of lines per day. The reasons for such a low productivity are in the time needed for verification of the system and long redesign cycles that come from the need of developing full system prototypes for the lack of appropriate virtual engineering methods and tools for embedded software. Embedded software is substantially different from traditional software for commercial and corporate applications: by virtue of being embedded in a surrounding system, the software must be able to continuously react to stimuli in the desired way, i.e., within bounds on timing, power consumed and cost. Verifying the correctness of the system requires that the model of the software be transformed to include information that involve physical quantities to retain only what is relevant to the task at hand. In traditional software systems, the abstraction process leaves out all the physical aspects of the systems as only the functional aspects of the code matter.

- *Integration Complexity.* A standard technique to deal with complexity is decomposing "top-down" the system into subsystems. This approach, which has been customarily adopted by the semiconductor industry for years, has limitation as a designer or a group of designers has to fully comprehend the entire system and to partition appropriately its various parts, a difficult task given the enormous complexity of today's systems. Hence, the future is one of developing systems by *composing* pieces that all or in part have already been pre-designed or designed independently by other design groups or even companies. This has been done routinely in vertical design chains for example in the transportation vertical, albeit in a heuristic and ad hoc way. The resulting lack of an overall understanding of the interplay of the sub-systems and of the difficulties encountered in integrating very complex parts causes system integration to become a nightmare in the system industry. For example, Jurgen Hubbert, then in charge of the Mercedes-Benz passenger car division, publicly stated in 2003: "*The industry is fighting to solve problems that are coming from electronics and companies that introduce new technologies face additional risks. We have experienced blackouts on our cockpit management and navigation command system and there have been problems with telephone connections and seat heating.*"

I believe that in today's environment this state is the rule for the leading system OEMs let them operate in the transportation domain, in multimedia systems, in communication, rather than the exception. The source of these problems is clearly the increased complexity but also the difficulty of the OEMs in managing the integration and maintenance process with subsystems that come from different suppliers who use different design methods, different software architecture, different hardware platforms, different (and often proprietary) Real-Time Operating Systems. Therefore, there is a need for standards in the software and hardware domains that will allow plug-and-play of subsystems and their implementation while the competitive advantage of an OEM will increasingly reside on novel and compelling functionalities.

I will present a methodology to cope with some of these problems and that can use hybrid system modeling. I will review how this methodology can be applied to the design of embedded controllers for the automotive industry. Finally I will present the application of the methodology and of hybrid systems to the design of wireless sensor networks in an industrial environment.