Is Embedded Software for Safety Critical Automotive Systems Really a Software Problem?

Alberto Sangiovanni Vincentelli

The Edgar L. and Harold H. Buttner Chair of EECS
University of California at Berkeley

Co-founder, CTA and Member of the Board
Cadence Design Systems, Inc.

Abstract:

As the electronic component of a car increases its strategic role in determining value added and market acceptance, the electronic design chain must be optimized in terms of efficiency and quality. Daimler-Chrysler executives expressed the dominance of electronics in future generation cars by underlying that 90% of all innovations will be in the embedded electronics. Most notably the driving experience will depend on embedded controllers of increasing sophistication. X-by-wire electronics will change dramatically the drivability of a vehicle. However, increased complexity has its drawbacks: Jurgen Hubbert of Mercedes Benz says: "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." Since these electronic subsystems are mostly driven by software and provided by Tier 1 suppliers who rely heavily on Tier 2 suppliers, it is clear that we need to concentrate on how to improve both the embedded software methodology and the communications among the design chain both in terms of specifications and validation.

Our take is that Embedded Software (ESW) design is one, albeit critical, aspect of the more general problem of Embedded System Design (ESD or just ES). ESD is about the implementation of a set of functionalities satisfying a number of constraints ranging from performance to cost, emissions, power consumption and weight. The choice of implementation architecture implies which functionality will be implemented as a hardware component or as software running on a programmable component. I believe that ESW should be developed by: 1) linking ESW upwards in the abstraction layers to system functionality; 2) linking ESW to the programmable platforms that support it thus providing the much needed means to verify whether the constraints posed on ES are met. To realize this approach, we have on one hand to develop formal techniques at the abstract level so that verification is started early and with the correct set of tools and methods. On the other hand, we have to think of ESW and hardware architecture in a unified and harmonious way. Platform-based design is a design methodology where re-use and programmability are the name of the game. Programmability here will also extend to hardware implementation with the advent of embedded FPGA that will allow a system designer to use another design trade-off point where functionalities can be allocated to programmable hardware blocks in addition to hardware and software.

I present a number of activities aimed at defining and building this methodology based on new environments and tools. These efforts are characterized by collaboration across company borders where suppliers and tool vendors team up to deliver better designs. These design methods for embedded controllers in a car are based on mathematical representations of the control problem,
rigorous determination of the quality of the control strategy and semi-automated generation of software implementations on distributed architectures.

In this talk, I will illustrate the main challenges and opportunities of the vertical design chain integration. In addition, we will present platform-based design as an important approach to meet the challenges and take advantage of the opportunities. The Metropolis environment will be described as a framework to sustain the methodology.