October 31, 2007

start:Optimize automotive 'supersystems'

Investigation of catastrophic specification and complex architecture paths completes, and gives competence to, electronics system engineering

By Graham Hellestrand,
Embedded Systems Technology

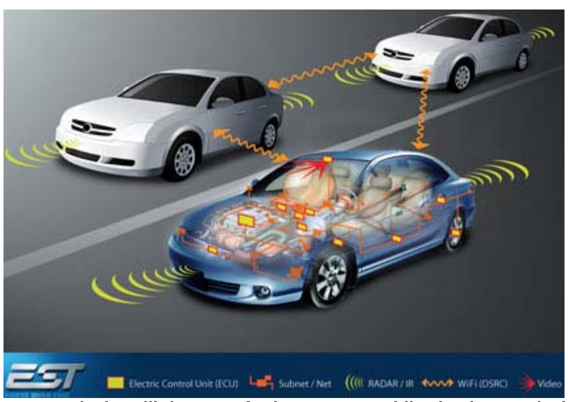
Modern vehicles (automobiles, trains, aircraft, or ships) are obviously complex. The vehicle control architectures (VCA) in luxury automobiles, for instance, contain 50-80 electronic control units (ECU), all sporting a microprocessor of various capabilities and all communicating over four to six networks. Yet no one is able to say whether there should be 80 or 10 ECUs interconnected by one or 10 networks in an optimal VCA. Not only that, but it also is not possible to exhaustively verify the real-time control systems of modern vehicles using formal methods, or by building physical prototypes and driving or flying them around for a year or two.

We have seen the results of these approaches to verification in the embarrassing stalling of luxury and hybrid cars on autobahns and highways due to improbable, but clearly inadequate, testing for faults occurring within, and increasingly between, ECUs in the vehicles.

The fact that such rough engineering has survived so long in the face of desirable optimization constraints involving cost, performance, reliability, fault tolerance, safety, and liability is rather remarkable in the cost-fixated transportation industries. These faults may have their origin in the specification process as well as design and manufacturing supply-chain processes.

Real-time complexity rising

The complexity of the empirical optimization and verification processes for supersystems (such as the VCA of an entire automobile or airplane) stems, in part, from the interaction of five technologies used to realize real-time control systems: Mechanical, electronic (digital and analog), computer system hardware, software, and communications. Another complexity factor is that the optimization (or verification, for that matter) of a single VCA must necessarily be performed in a realistic context—within that of *other* vehicles VCAs as well as traffic infrastructure (below). Remarkable as it may seem, the lack of optimization as a specification imperative is all too obvious in the automotive industry.



Real-time systems complexity will rise even further as automobiles begin to actively communicate with each other, as well as the infrastructure, to share information on traffic flow, etc.

Get it together

Two processes that should be twinned in systems engineering are specification and architecture with verification and validation. Real-time control system specification and architecture is a discipline in its own right, but it is only beginning to be used by the early adopter companies to drive their design processes.

At the other end of the conventional and widely-adopted V engineering process (serial design, test, and manufacturing process)—systems integration—lies a very thorny issue: The failure of a control system detected during the verification of an integrated system.

Failures that occur late in the engineering process are always expensive to repair since they require complex, and likely lengthy, system diagnosis, followed by either radical system redesign or cancellation at the point of peak expenditure in the project. It is the job of empirical system architecture to flag and remedy problems, including the potentially catastrophic, in the initial phase of the systems engineering process. Indeed, until catastrophic specification and architecture paths are investigated, the empirical systems

engineering process is incompetent and incomplete. The conventional V process, perversely, often seems to be the reverse of a competent systems engineering process.
The alternative approach to the *post facto* verification that is built into the V design process is to specify systems as executable architectures. And, in parallel, to build a verification process based on simulation of use-case scenarios from which test cases are derived. As the specifications are mapped iteratively to models of physical systems, so test cases that address the additional structure, function, and timing of the mapped models are accreted to those already developed.

This methodology is a highly efficient concurrent architecture and verification process enabled by model-based design together with a coherent process of mapping specification and architecture to physically realizable design. There remains a vestigial tail of the V process in the physical (manufacturing) supply chain, but with a twist—the validation of manufactured subsystems and systems is achieved when all of the applicable tests generated during concurrent design and verification are applied, and the responses conform to the validation responses constructed during specification and architecture.

Recent breakthroughs in system simulation that use cheap and available multi-core host computers connected by high performance (mainly, low latency rather than high throughput) interconnect fabrics have enabled the simulation of models of very large systems with high accuracy and performance. These modeled systems can be subjected to exhaustive experimentation, using design of experiments standard methodology and multi-variate statistical techniques to simplify the analysis, to yield demonstrably correct and optimal architectures before being reduced to physical implementation. This procedure is the reverse of the current practice of build, then verify.

The relationship between supply chains and system architectures is intimate. An abstract architecture defines, in part, the structural relationships between the entities (subsystems and components) that will be assembled into a system. This approach applies whether the relationships are based on composition (vertical—forming hierarchies; horizontal—forming sets), communication interfaces, or interconnect fabrics. The building of the entities and their (progressive) integration into a system—while observing the rules of composition, communication and interconnection—is what a manufacturing supply chain does.

An architecture also defines the function and timing of:

- Computations performed within each entity (that is, the control algorithms in the electronic control unit of a car, for instance)
- Communications between entities
- The interconnect fabrics.

The mechanisms for delivering the functional and timing specification dictated by the abstract architecture are the business of the design supply chain.

The design and manufacturing supply chains are often different, matching the strategic advantages of potential suppliers in either chain. However, the degree of concurrent activity and the sequential constraints in the supply chains are dictated by the system architecture. It is in this sense that an optimized system architecture will have a profound effect on the efficiency and effectiveness of the supply chain.

In addition, the verification process that is twinned with the process of mapping specification/architecture to proximal physical models in the design supply chain produces the validation test required by each participant on the manufacturing supply chain (at each stage of entity creation and entity integration). In this way, the design and manufacturing supply chains are like the intertwined strands of a DNA molecule—the bridges between the two standards being the validation tests that are created as part of the design supply chain verification process and used in the manufacturing supply chain.

***Graham Hellestran** is a founder and CEO of Embedded Systems Technology and a member of the board of directors of National ICT Australia Ltd. (NICTA). Hellestrand was founder and CEO of VaST Systems Technology from 1997 to 2004.*