Safety & Architecture: Engineering Systems and Systems of Systems

Graham Hellestrand, Embedded Systems Technology, Inc. and University of NSW, Australia
Casey Alford and Neville Clark, Embedded Systems Technology, Inc.
(g.hellestrand@essetek.com)

Overview and Focus: The two foci of this submission are:

i. **Component-based engineering versus architecture driven engineering**: Component-based design is a methodology adopted from production engineering where the physical constraints of assembling physical systems – basically, compatibility of interfaces – are either prescribed by a preset system architecture or prescribe a restricted set of architectures for a system. Neither prescribed outcome is attractive when: (i) high-level, executable functional specifications can drive architectural exploration and optimization that result in a small set of architectures – composed of parameterized hierarchies and oligarchies of sub-architectures (subsystems of functionals) - that meet requirements and satisfy objective functions; and (ii) *component* functionals can be translated directly to software or hardware *components* with complying interfaces. This is the essence of the technology and methodology embodied in model-based, architectural driven engineering. The component based design methodology can yield dependable and secure cyber-physical systems backed up with elegant demonstrations of preservation of static properties under composition. The architecture driven engineering technologies systematically and efficiently enable both static and dynamic – for instance, *collision avoidance*, *minimum emissions and energy consumption* - constraints to be checked during the development process. Once identified, compatible pre-existing components and/or subsystems can be used to build the physical system from the model. In production, 6-Ω tolerances are a feature of manufacturing quality that underlies a quantification of dependability, as well as other attributes.

ii. **Safety and Functional Safety is a Requirement:** The ISO 26262 standard and the IEC 61508 standard from which it is derived both have dependencies on the ISO 9000 series of standards in that there is assumed to be a rigorous documentation trail underpinning the engineering of safe systems. Functional Safety involves an active function that monitors a system to ensure that it will, with a risk that is as low as possible, not *physically injure or damage the health of people directly, or indirectly as a result of damage to property or the environment.* [*IEC 61508 Introduction*]. All aspects of automotive vehicle safety are within the ambit of the functional safety prescriptions which require the consideration of the entire system and the environment in which the system operates (including traffic). At present, no OEM or Tier1 appears to be able to conform to the environmental requirements of the standard – in particular, with regard to the verification of vehicles in traffic.
Safety, in contrast with Functional Safety, is a broader term that admits – we would say requires - the incorporation of safety into systems and systems of systems, as a Requirement.

Systems and Systems of Systems

Architecture based design is predicated on modelling and simulation. It starts with a set of executable specifications (models) derived from the Requirement document that incorporates the usual defining elements of distributed real-time control systems that govern plant, together with Safety requirements, scenarios, use cases and system tests. At present, there is no automatic means of extracting a model from its Requirement document, even though work is progressing towards this goal. So executable specifications (models) are built manually from Requirements. Ideally, such models are functional but include accurate timing and have the form of systems of functionals – differential & integral equations, state machines, data-flow machines, etc. – linked together, for instance, using Simulink and other modelling notations that enable the representation of computational/control and physical plant sites and the communications links between them. Multiple specification models can be constructed from one Requirement document and they are equivalent if each satisfies all the scenarios, use cases and test cases of the Requirement. Typically, specification models are parameterized to enable various functional, structural and timing aspects of the system to be exposed and varied, and the effects of these variations explored under simulation. Hierarchical and oligarchical (also called parallel) composition, together with concurrent communication, is fundamental in specifying and modelling systems of systems, such as, high fidelity vehicles in traffic. The careful parameterization of models enables the creation of families of control system architectures, which may be highly relevant across model years and for planned migration through models, and their assessment in terms of fitness for purpose. In such scenarios, reuse is a desirable characteristic since if the original module was built under the dictates, say, of the ISO 26262 or IEC 61508 standard, the reused modules will still exhibit the functional safety requirement.

The construction of very large architectural spaces is effected by iterating through the levels (values) of each set of parameters in each specification model. For example, a model with 6 independent parameters (dimensions) having, say, 10,11,12,15, 20 and 5 discrete (value) levels, will generate a 6-dimensional architectural space with 19.8 million points. This enables extensive experimentation to occur when optimizing systems to satisfy multiple objective functions. We use Design of Experiments and significant pre-experimentation sampling of the architecture space to address the curse of high dimensionality when running large-scale optimization studies. There are a number of open DoE packages – for example, the R-Project statistical package, as well as proprietary packages such as the MathWorks Optimization Toolkit.

In the automotive domain, high fidelity vehicle models are calibrated against physical systems. Such real-time, distributed system models incorporate advanced engine, transmission, braking, steering and suspension plant models each controlled via networked electronic control unit (ECU) models executing real – often production - software. Traffic systems of vehicle, infrastructure, traffic controller, and traffic management system models are formed into large-scale distributed systems that model significant urban areas in large cities. The studying of such systems is fundamental in the quest to find optimal solutions to safety (collision avoidance and collision damage mitigation), energy consumption, gaseous & particulate emissions, infrastructure utilization, economic control of congestion, etc. in large grid-locked cities where the death toll due to emissions exceeds that of collisions and other crashes. One promising avenue of investigation is cooperative control of many vehicles. To enable this high fidelity wireless communication (such as, DSRC.802.11p) and sensor (such as, radar, lidar, IR, etc.) models are required in each vehicle model to support high accuracy communication amongst vehicles and infrastructure such as road-side units. Such complex, high fidelity traffic systems demand high computation and communication - typically large multi-core - infrastructure to simulate quickly and with high functional and timing accuracy.

Safety and Functional Safety:

From the perspective of architecture driven engineering, there is no such concept as passive safety – the control of emissions, fuel and the mitigation of damage in collisions is primarily an active exercise very likely coordinated amongst a group of proximal vehicles. Automotive Safety and Functional Safety only make sense in the context of place, time, situation and traffic – it is a system of systems issue and cannot be specified or engineered into a vehicle in isolation. It is difficult to know how safety – being dynamic – can be guaranteed using component based design; it is an intrinsic part of architectural driven engineering.

The Short Term: The system described above exists and has been in use in a number of customer locations. This is not an Open System but it does admit the incorporation of any open and/or proprietary model, simulator or tool. For example:

i. For physical plant, sensors and actuators, it supports the incorporation of mechanical, electrical, thermodynamic, fluidic models: Active interfaces to Saber, Simulink, Simplorer and C/C++ modles.

ii. For electronic control units (ECUs) - processors, peripherals, sensors, buses and networks: Active interfaces to C/C++, SystemC, Synopsys/VaST, ARM and SILS – Software in the Loop – models.

iii. Full vehicle dynamics models: An active interface exists for CarSim models.
  - CarSim models have been augmented with high fidelity engine, transmission, braking and suspension models and their respective controllers.

iv. Roadway infrastructure plus traffic light controller models are already incorporated into the system.
  - We will interface a Paramics-style simulator and visualization system later this year.

v. The development, testing & verification of software using editors, compilers, and debuggers is supported, including the concurrent use of such tools across multiple ECUs.

The Longer Term: For the longer term, architecture driven engineering – including safety, failure tolerance and optimization over high dimensional spaces is the focus of the effort. Even though the system can be used for building large-scale models and simulations of any distributed real-time systems, the foc0us will remain on automotive systems.
Some longer term efforts will include:

1. **Automotive Secure High Confidence Platforms**:
   A. **Architecture-based design**: Specify and investigate, parameterized systems and systems of systems in terms of, say: (a) architectural patterns - for example, the use of fault tolerant architectures in safety applications, (b) technological change - including the expected change in clock-speed over the next decade,(c) interconnect structure – including single and multiple networks of various types, (d) economic factors – including those that affect health, safety, provision of private vs public transport, etc., and (e) sophisticated driver behaviours.
   B. Another long-term effort will be to investigate the creation of families of control system architectures, which may be migrated across model years and through model families. The particular outcomes that will be interesting include: degree of reuse this strategy will enable, the influence on the supply chain, the effect on costs, the cost of faulty projection 5-10 years in the future.
2. **Safety Critical Design Process**
   A. Continue to develop methodologies and technologies to enhance architecture-based design and to investigate the optimization of various objective functions – safety, energy, emissions, infrastructure - associated with large-scale, high fidelity models of traffic in dense urban areas.
   B. Develop a methodology to use the Design of Experiment to verify that the behaviour of large-scale systems complies with its specification. This involves identifying unexpected and unexplained behaviours of systems during simulation including the failure to trigger functional safety mitigation actions in the event of a failure.

Biography:

Graham Hellestrand is Founder and CEO of Embedded Systems Technology, Inc. – his 3rd start-up and the 2nd in Silicon Valley. He is a member of the Australian Government's Information Technology Industry Innovation Council and was Director of National ICT Australia Ltd (NICTA) 2006-2010. He is Emeritus Professor of Computer Science and Engineering, University of NSW and holds BSc(Hons), PhD and Exec MBA degrees from that university and an MBA, University of Sydney. He is a Fellow of the IEEE and Fellow of the Institution of Engineers, Australia. He has published in excess of 100 papers in international conferences and journals and is the principal author of two patents. He has held a number of Board position in IEEE CAS between 1994 & 2010. He has lived in Silicon Valley for the past 13 years.