Engineering Safe Autonomous Mobile Systems of Systems
Using Specification (Model) based Systems Architecture & Engineering

Graham R. Hellestrand
Embedded Systems Technology, Inc., San Carlos, California, USA
g.hellestrand@essetek.com

***Abstract*—Engineering safe, complex real-time systems is challenging. Engineering mobile systems of systems that are safe and possibly autonomous, requires considerable support from competent specification based architecture, model-based design processes and concomitant large-scale, heterogeneous simulation capabilities. Safety – the dominatrix of autonomy - is determined by requirements that then propagate through the specification, architecture, design, verification, validation and calibration phases of the real-time engineering process. In real-time systems, time is a 1st class, functional property of the system. The paper describes a specification-based architecture for the engineering of safe mobile system of systems and the modeling and simulation technology required to produce them.**

Keywords – specification based architecture; safe mobile systems of systems; safety and autonomy; large-scale, real-time, distributed control systems; heterogeneous, systems architecture & engineering; empirical optimization of systems of systems; productionquality model-based design.

I. INTRODUCTION

Actively controlling the safety and predictability of semi-autonomous and autonomous mobile systems is the dominant factor governing their engineering. In short, whilst fulfilling their purpose, the civil objective of mobile systems is that they should not negligently kill or harm their riders, pedestrians or animals, or impair the health of living creatures with particulate and gaseous emissions. If we add, avoiding harm to the environment and damage to property – urban, regional and remote – then this is simply the goals of *Functional Safety* as defined in the IEC 61508 standard [1] and the derivative ISO 26262 Road Vehicles – Functional Safety standard that relates specifically to the electrical and/or electronic systems (apart from active and passive safety systems) installed in production road vehicles of less than 3,500 Kg [2].

The conundrum is how to go about building, verified safe vehicles. The problems come starkly into focus when one considers - Does safety include what can be done to actively (as well as passively) avoid death and mitigate injury during collisions in traffic? Most would hope that this might be the case. But how does an industry verify such levels of safety without killing people? The principal drivers of safety in mobile systems of systems include: excellent safety-driven requirements; optimizing efficacy over objectives such as minimizing complexity, risk and cost, while maximizing verification coverage; and doing it at a price that customers will value and pay for. The paper addresses these issues.

II. SYSTEMS OF SYSTEMS AND SAFETY

Estimating the risk associated with guaranteeing the functional safety of systems – doing no harm to humans, animals & property – is quite different from estimating the risk associated with guaranteeing the functional safety of components or subsystems that constitute the real-time connected control and plant of a system. Inherently, subsystems and components live within the protective cocoon of a defined system where, relatively, a great deal is known about the timing and functioning of the interconnected components and sub-systems. Systems, however, exist in *external* environments and not within the control domain of the system itself, which brings into play, the issue of estimating the risk associated with both known unknowns and unknown unknowns – the latter having an incomputable level of risk. Systems of systems (such as traffic control systems), attempt to mitigate the risk for systems (automobiles, pedestrians, property) by defining another cocoon of control; the external risks remain but some of the unknowns are now defined and hence the risk is reduced for the systems within.

The bottom-up composing of systems from functions, as is the predominant mode of designing vehicles today, is hazardous. It is akin to designing a forest from its trees, or worse trying to make a car in traffic safe just by making its components safe. This mode of design appears to have been driven by a sense of overwhelming complexity as ECUs and software proliferated throughout vehicles' control systems – powertrain, chassis, body - and infotainment. As the AUTOSAR standard states, its 1st motivating issue is the *Management of E/E complexity associated with growth in functional scope* [3].

A. Specification-Based Architecture

This paper advocates the use of Specification-Based Architecture (SBA) in the engineering of systems and systems of systems. It differs from model-based design (MBD) and platform-based design (PBD) [4] in that it is a top-down approach that uses both function and timing during specification and is unconcerned with realization prior to defining an architecture – or set of architectures – having causality, deterministic behaviour, required latency and bandwidth characteristics for both computation and communication. In contrast, MBD is fundamentally driven by components that are bottom-up composed into sub-system hierarchies. There are many toolsets that support model-based design that utilize many notations from differential equations, to finite state machines, to software code and hardware

components. Platform Based Design, as described in [4] is an iterative process that is partly top-down (function) and partly-bottom up (architecture – platform as a library of selectable components).

SBA typically uses differential or integral equations augmented with functional and structural parameterization. These physics models of systems are deemed to capture the dynamic state and timing of systems. The parameterizations are capable of generating large architectural spaces that admit to searching directed by objective functions. For an operational semantic of such systems see [5]. The advantage of SBAs is that they capture computational characteristics, timing characteristics, signal characteristics and communication characteristics as 1st class entities. Refinements of SBAs can be done in terms of more detailed systems of differential or integral equations. SBAs are specifications since they are derived directly from the requirements and are executable (using numerical differentiation and integration methods) and hence capture both structure and dynamic function and timing.

B. Safety in Systems and Systems of Systems

For automotive systems, which should be designed and constructed using safe engineering processes, the analysis of faults that cause errors that lead to failures that produce hazards that result in a crash is the typical chain of successive causes and effects that is analyzed – it is inherently bottom-up. Since we are starting with an executable specification of a system the identification of hazards – system states or sets of conditions that, together with a particular set of worst-case environmental conditions, will lead to a crash [6] – are primarily our concern and hence the starting point of safety in a top-down process. As the process of refinement from requirements→specification→architecture→design→calibration→realization→validation progresses, so will the identification of potential crashes (safety constraints) as part of requirements, hazards as part of specification and architecture, failure and errors as part of design, errors and faults as part of realization. Systems of systems must have safety built into the requirements that then dictate the construction of the specification. Note that verification is progressive and occurs at every stage of the process. Verification uses scenarios to test specifications and a mixture of test cases and scenarios to test artefacts at the architecture, design and realization stages.

A vehicle and its control system are typically regarded as a relatively complex system. A cohort of cooperating, communicating vehicles produces an orderly mobile system of systems and it is considerably more complex than a vehicle. When traffic control and infrastructure constraints, expected in an urban environment, are included as part of the system of systems, it becomes a city transport system with high potential for supporting adaptive and autonomous behaviour. Autonomous mobile systems are probably the most interesting and challenging inanimate systems ever conceived and engineered. Such systems drive, fly, swim and float around in space –with an increasing degree of autonomy, though, at present, largely under *adult* supervision. As magical as such systems are, they also engender, for living things, inventive ways to die, be maimed or just damaged. Humans are strangely tolerant of such outcomes and often intentionally do not design for, nor verify, the control behaviour of mobile, let alone autonomous, systems when operating in the death zone – that is during a crash.

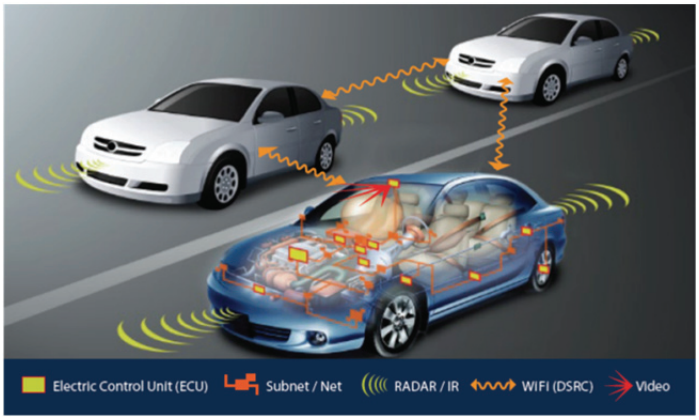


Figure 1. A Modest System of Communicating Vehicles Cooperating as a System – perhaps Autonomously.

C. Identifying Hazards in Systems and Systems of Systems

Figures 1 & 2 help identify potential crashes at the system of systems level. The importance of performing this analysis and embedding the results in the requirements is to build safety into that specification. Building safety in from the bottom-up, by considering faults in components, or fault-trees from the vehicle down to its components only cover a subset of hazards and make little sense for the overall safety of a vehicle operating in an urban traffic zone. A typical potential crash that arises at the system of systems level is 2 vehicles involved in a side-impact crash; the hazards include: intersection control failure, driver inattention, vehicle brake failure.

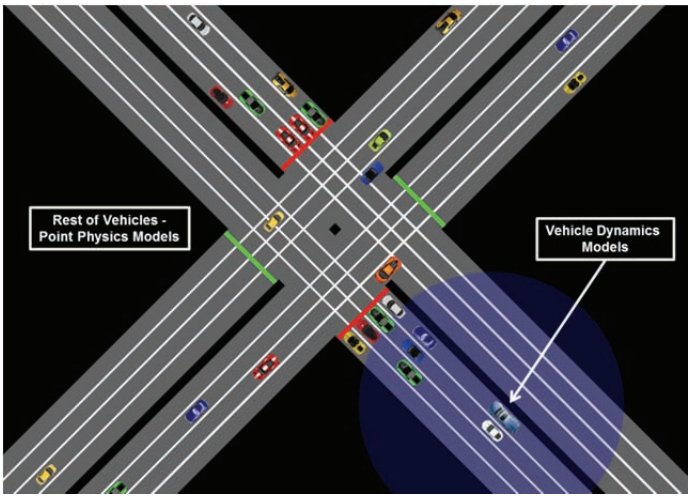


Figure 2. A Many Vehicle Systems of Systems Illustrates the Potential Crashes and the Hazards that may lead to a Crash

In summary, without knowing the details of the engineering design there is already a large set of hazards that help define equations that build into the architecture the safety requirements. In addition, the same specifications define the required test scenarios that will become part of the verification suite to be used through all stages of the engineering process. As additional detail is added during the reduction of a specification to engineering realization scenarios and tests to check these details are added to the verification suites. It is during this reduction process that faults, errors, and failures are identified and mechanisms incorporated into the engineering design to resolve these issues or to mitigate their effects. For safety critical subsystems, this may involve triple multiply redundant design.

D. Validation and Calibration via SBA

In this environment, validation is the demonstration that a physical system and its executable specification pass the same tests and the validated artifact is suited to perform according to its requirements. The caveat here is that, even though differential and integral equations are excellent for modeling physical systems, they do not model them precisely. In part, this is due to such models typically not accounting precisely for the characteristics of physical systems at the minute, atomic and sub-atomic levels. So they are approximations to reality for plant involving mechanical, thermodynamic, aerodynamic, and fluid flow characteristics. On the other hand, the control systems specified with the same mathematics work well since they are only to have effect at the macro level.

Calibration via SBA encounters the same issues as validation. Due to the semantic and structural mismatch of the SBA and the physical system, calibration does require an iterative approach in which both the physical system (which had been calibrated directly from the SBA), and its SBA are fine tuned to match under the authority of expert test drivers. However, calibration in an SBA-driven process should limit the number of required *mule* vehicles in any new production run to a handful rather than many tens.

III. Architecting Optimal Vehicles from Executable Specifications

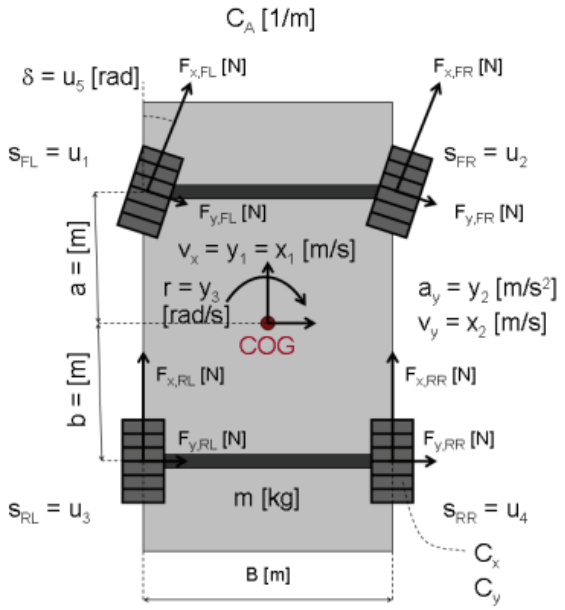
At the most abstract level, a vehicle is a set of equations that define the moving, acceleration, deceleration, turning, communicating and cooperating characteristics of car and driver. The models range from the simple, called point physics (see Fig. 2) or ballistic models, to the very complex, known as vehicle dynamics models (VDMs – see Figure 3) [7], [8].

The point physics models are essentially constructed as a set of rules with a simple, but still sophisticated, driver model, such as the Gipps model [9]. The VDMs are constructed from systems of equations - typically differential, differential algebraic or integral equations - that model: automobile layout and geometry, mass distribution, aerodynamics, steering, suspension, traction (including tires and road surfaces), motion and driver models. The VDM models are typically augmented with simpler models – often table lookup - of complex vehicle plant such as engine and transmission. The physical architecture may or may not be well represented in these sets of equations.

The VDMs form a mathematical specification of a vehicle based primarily on the classical mechanics of multi-body systems. It is also worth noting that the equations are likely to contain control functions – using the same mathematics – that, for instance, might control the stability of a vehicle.

Of major importance is that the mathematical specification of the physics of a vehicle's dynamics is able to be simulated using numerical differentiation and integration techniques. This means we can begin architecting a vehicle using an executable specification. Of singular importance is that the vehicle models are able to be parameterized. Indeed, VDMs such as CarSim and TruckSim [10] have many hundreds of parameters that affect both function and structure. These parameters form an architectural space defined by the sets of equations with axes defined by the parameters. Figure 4, below, provides a 2-D representation of a multi-dimensioned space with many architectural points.

http://www.dspace.com/en/inc/home/products/sw/automotive_simulation_models.cfm



http://www.mathworks.com/help/ident/examples/xxxfiguresvehicle.png

Figure 3. Vehicle Dynamics Models

A VDM with hundreds of parameters forms an architectural space of hundreds of dimensions, in which each dimension might have, say, between 2-100 values. This space is intractably large to search exhaustively for optimum architectures. Many parameters of such systems are likely to be highly correlated, in which case, the correlated parameters can be aliased and represented by a single parameter. However, even an 8 dimensioned space with each dimension having 10 values contains 100 million discrete architectures. There are many and varied techniques for searching in such large spaces some of them random (for example, Monte Carlo), others semi-random (for example, simulated annealing and genetic algorithms), others highly directed using methods of steepest ascent/descent defined on the objective functions over the space (for example, Design of Experiments), and yet others that are algorithmic. All methods, with the exception of random search, attempt to reduce the search time and effort.

The important point here is we are performing architectural searches in a high-dimensioned space defined by a set of abstract, mathematical functions. The formulation of the sets of differential or integral functions may indeed have embedded in them physical characteristics of architectures since they have been formulated from the classical physics of mechanics and aerodynamics. In essence, we can find a set of optimal architectures of a vehicle defined using abstract mathematics. We have also incorporated the requirements driven by safety analysis to ensure that safety has been incorporated as a fundamental architectural objective. An excellent exposition of safety-driven design is given in [11].

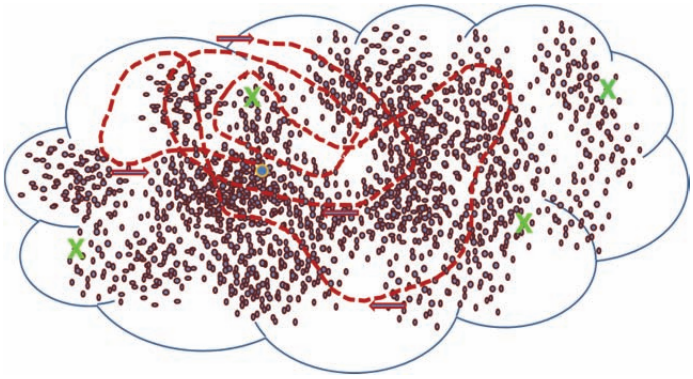


Figure 4. A Large Dimensioned Space with Paths (red) through it Representing Directed Searches Attempting to Find Optimal Architectures Defined by Sets of Objective Functions. The X's Represent Historically Driven or Randomly Selected Architectures that are Unlikely to be Optimal.

IV. Augmenting VDMs and Reducing SBAs to Engineering Practice

Vehicle dynamics models typically do not contain detailed models of powertrains (engines, transmissions, differential, axels) braking systems, steering systems, active suspension systems, and drivers. Although they, together with point-physics models, are the foundation of automotive systems of systems work, they are insufficient to drive the reduction to engineering practice.

A. Augmenting VDMs

VDMs need to be augmented to form SBAs where the VDM's subsystems of interest, such as the engine and its control, are replaced by high fidelity specifications. Typically, high fidelity models of engines, transmission, brakes, etc. are built by the automotive vehicle manufacturers – called in the industry, Original Equipment Manufacturers (OEMs), or automotive subsystem suppliers – called, Tier 1s. Primarily, these models are built using continuous domain modeling notations and are themselves executable (simulatable) specifications.

A typical model of plant + control is given in Figure 5. In fact, there is more involved than just plant + control. The plant models have associated with them both actuators, which effect changes in the plant model as ordered by the control, and sensors, which provide status (state) information about the plant back to the control. There is also a cause and effect chain operating: *Control → Actuators → Plant → Sensors → Control*. This is classic feedback control that has the ability to order future actions depending on the state of the plant. In Figure 5, synchronization mechanisms are required to (i) manage the input to the plant from the external world together and from the control, and (ii) to coordinate the sensor information back to the control as well as to the external world. Even without the interactions of the plant + control subsystem with the external world, synchronization of control and the plant interactions is required.

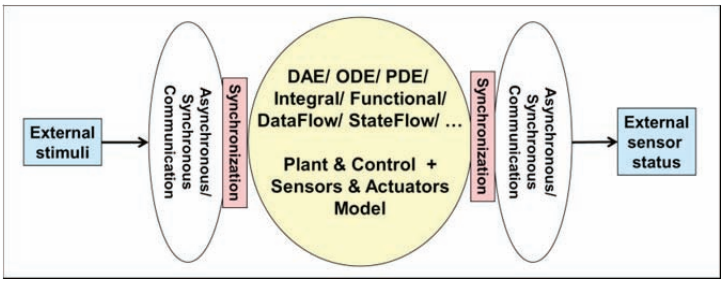


Figure 5. Executable Specification of Plant and Control

All of the plant + control subsystems that are required to augment a VDM to form a high fidelity SBA of a vehicle have similar structure, and communication and synchronization requirements. It is this generality that allows a standard process to be employed to reduce such executable specification to proximal physical models – that is, high fidelity models of physical plant and its control) and then to engineering artefacts (the real plant and control).

B. Reducing SBAs to Engineering Practice

A modern physical automobile incorporates a real-time control system containing 5-20 networked electronic control units (ECUs) executing perhaps several million lines of software. The control system governs the operation of a similar number

of physical plant in such a way that they coordinate the movement of the vehicle under the control of a supervisor (driver) – whether human or robotic. The ECU's communicate via multiple wired networks (such as, CAN, FlexRay, LIN, Ethernet) and the vehicles communicate via multiple wireless networks (such as, mobile WiFi, 3G). Communicating vehicles can form ephemeral mobile networks with the capability of exchanging information and synchronizing activities at times when they are in close proximity (approx.. 400m). This capability can be used adventitiously to form, control and disband cooperating cohorts of vehicles with the objectives of increasing safety, reducing fuel consumption and emissions as traffic circumstances permit. Such vehicles have high potential for being controlled autonomously.

How do we get to a vehicle from an SBA? There is a 5 step process described in the following sub-sections. Additional details may be found in [12].

1) Disjointing

There are two mechanisms for decomposing a subsystem as depicted in Figure 5. The 1st type – disjointing into subsystems - is shown in Figure 6.

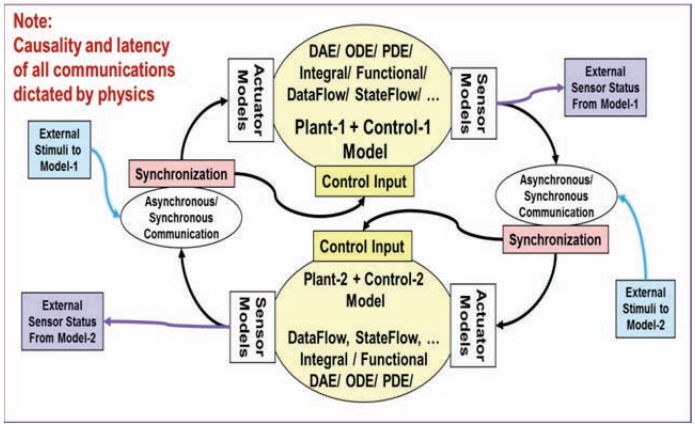


Figure 6. Disjointing into Integral Subsystems

We have called the process of extracting/decomposing subsystems from another subsystem *disjointing* since it better reflects the process of separating out sets of equations that reflect each physical subsystems. And then establishing the appropriate re-connections of the disjointed subsystems.

The 2nd type of disjointing process is shown in Figure 7 in which the disjointing process uncouples the embedded plant and control subsystems and inserts the required synchronization to implement the feedback control between the plant and the controller. The synchronization also enables external inputs and outputs to be coordinated appropriately with the feedback control.

The 2 types of disjointing process have produced stand-alone subsystems that reflect some partitioning of the original SBA and/or an extraction of the plant and controller parts of subsystems, perhaps already disjointed from the SBA.

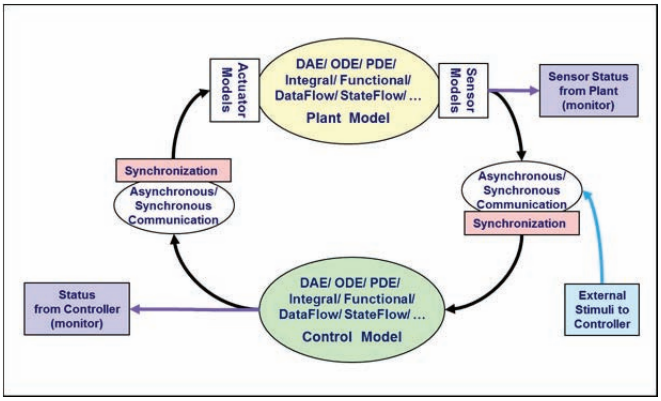


Figure 7. Disjointing into Plant–Control Pair

2) Reconnecting the Disjointed SBA

It is now possible to reconnect the disjointed subsystems as network connected subsystems that can cooperate together and potentially run in parallel, thereby increasing throughput in the system. Figure 8 shows the connection of each subsystems to a common network. In order to accomplish this connection we need to augment each of the plant-control pairs with a network connection that, at the minimum, synchronizes each control subsystem with the network to which it is attached and enables transmission and reception of messages.

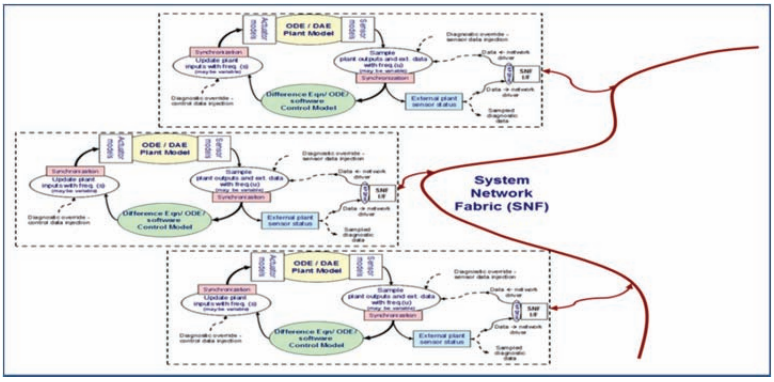


Figure 8. Three Plant-Control Pairs Connected via a Network

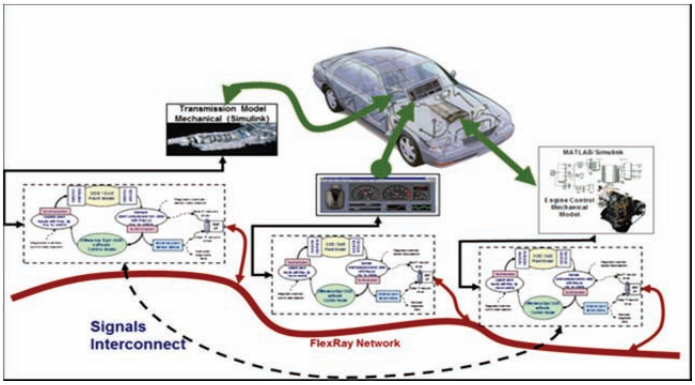


Figure 9. Distributed Control Reintegrated into the VDM (new SBA)

3) Distributed Plant-Control Reintegrated into the VDM

Figure 9 shows the reintegration of the 3 network connected plant-control pairs into the VDM. Now this version of the vehicle SBA and the executed concurrently with the original SBA and the outputs compared. It the outputs are not identical in function value and timing with the original SBA, either the modeling of the original SBA was inadequate or the conversion of the SBA to a VDM + more desirable plant-control subsystems is in error. Either way the faults causing the errors need to be determined and rectified.

4) Mapping Control Equations to Hardware or to Software and Executing it using an ECU (or modeled ECU)

The last but one step in reducing an SBA to engineering practice is mapping the control of the plant-control pairs to hardware (finite state machine, perhaps) or to software executing on an ECU. The ECU is required to have sufficient calibre to perform control work with the same function and timing as that effected by the original mathematical control equations.

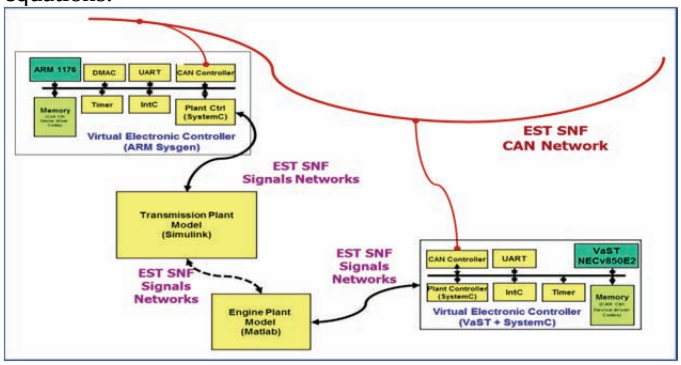


Figure 10. Control Mapped as Software to ECU Controller and 2 Controllers Connected via CAN Network

Figure 10 shows an engine-control and a transmission-control subsystem in which the control has been mapped to software that can be executed on an ECU.

5) The Final SBA Engineering Artefact

The final engineering artefact (see Figure 11) results from the process steps described above followed by reintegrating the hardware or software + ECU produced in process Step 4.

Once again this engineering version of the SBA can be compared with either of the artefact produced in Step 3 or the original SBA – since both of those have been shown to have identical function and timing outputs over the test cases and scenarios created though process Steps 1-4. If the results of the engineering SBA are not comparable, then the errors associated with mapping the control to hardware or software + ECU need to be identified and fixed.

The testing of an SBA, which is an accurate VDM integrated with accurate plant and control models, is capable of running test scenarios, such as the Japanese 10-15 road test. The instrumentation associated with an engineering SBA is capable of recording every event in every model being simulated. The use of scenarios, along with the test cases required of each additional mechanism introduced in the process of mapping and SBA to an engineering artifact, for the regression test used in verification.

V. Executing SBAs and Engineering SBAs

The SBA engineering process has been designed to provide efficient simulation for its executable specifications and models of mapped SBA engineering artefacts. The objective is for high fidelity models executed using high performance simulation on multi-core, multi-computer engines to unleash a heightened level of both engineering creativity and productivity.

This objective cannot be approached via the manual software and physical electronics engineering technology of today's production engineering processes.

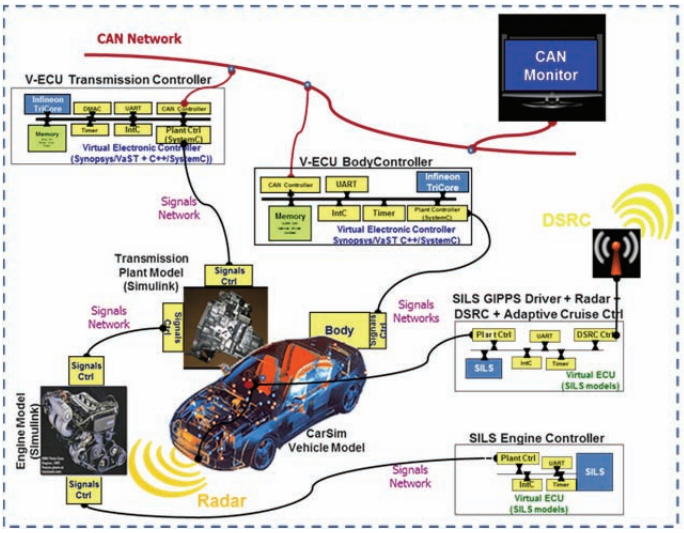


Figure 11. Final SBA Engineering Artefact

Figure 12 shows the mapping of 2 complex subsystems models consisting of 6 Virtual ECUs, 5 Plant models, 2 Monitors and 2 Network models (CAN Bus and FlexRay Network) mapped to 16 virtual cores (threads) over 2 Intel Core i7-based computers interconnected via a giga-bit Ethernet.

The subsystem and network models are automatically mapped to the cores and computers of the execution engine. The simulation performance for reasonably balanced computation and communication loads shows approximately a linear speedup wrt the number of models in system or system of systems. This is due to each computationally demanding model being allocated its own core and the high performance network, high fidelity models being able to perform at 1.5 – 2.5 million transactions per second. The network models are functionally and timing accurate and include arbitration where appropriate. The network models are themselves distributed and have no central control.

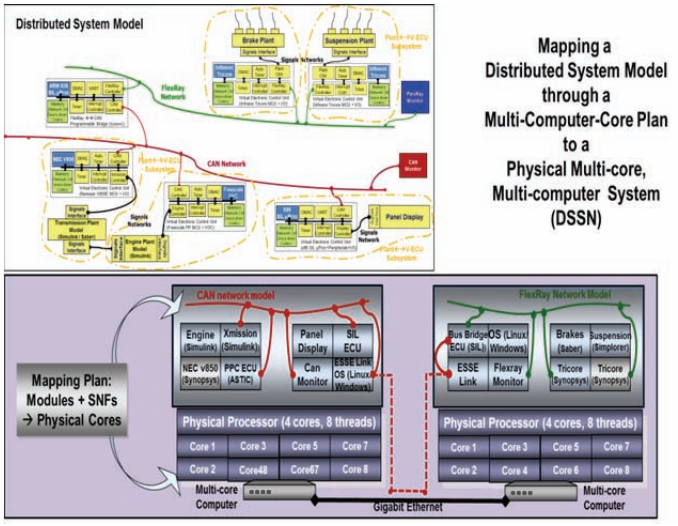


Figure 12. The Mutli-Core, Multi-Computer Distributed Simulation Engine

When point physics vehicle models are used in traffic simulations, up to 500 point physics models – complete with driver models and IEEE802.11p (DSRC) radio + IEEE1609.4 (MAC) models – are allocated per core of the execution engine. Large traffic and safety simulations (up to 2,000 vehicle models running on a 3D track 48 Km long) have been run that included a mix of high fidelity SBA vehicle models mixed with point physics vehicle models. One such study investigated the safety of the DSRC (Mobile WiFi) standards. The results indicate that with a single radio DSRC capability in each vehicle, when concentrations of vehicles within reception range of a radio reaches about 100, it may take more than 10 seconds to receive Basic Safety Messages (BSM) from all vehicles. In this study, there was not good correlation between receiving a time to receive a BSM and how close a transmitting vehicle was to the receiving vehicle. At ~60 mph, 10 seconds is equivalent to a travel distance of ~900 feet. Our conclusion is that the current standard DSRC system is not safe for use in safety applications associated with vehicles in many urban traffic situations.

VI. SUMMARY

The paper has described a Specification-based Architecture process in which the specification is derived directly from requirements in which safety constraints have been embedded. The safety requirements ripple through each step of the engineering process and are included as part of the scenario and test case verification. The paper also introduced an SBA and model-based design process for system of systems engineering as well as the simulation technology required to use it. It has discussed and demonstrated the ability to model heterogeneous systems, consisting of discrete and continuous models, with high-fidelity and high simulation performance.

Architecture-driven, model-based technology, when deployed as a foundation of engineering, extends the reach of engineers from the daily design of real-time systems engineering into the realm of the design of apparently ambiguous control underlying active supervisory systems, such as that often involved in collision avoidance decision making. In event of the failure of supervisory systems, the system enables an empirical investigation of the shadowy causes of, and extraordinary engineering of, systems that mitigate the frequency of death and severity of injury during crashes.
The modeling of the physics of vehicles in crashes is not part of the current system but is worthy of investigation.

The fidelity of the models of vehicles and traffic enable the empirical investigation of autonomous driving in traffic- see Figures 1 & 2. Including examining the situations in which drivers might be killed. Such investigations are not possible using physical vehicles and it is an area where this scalable, high fidelity modeling and high performance simulation offers unparalleled advantage.

The same basic technology that underpins empirical model-based investigation, delivers the capability to optimize systems and achieve outcomes that satisfy multiple objective functions.

ACKNOWLEDGMENTS

The author would like to thank N.A. Clark, C.A. Alford and J.R. Torossian for constructing an extra-ordinary modeling and simulation system and for contributing to the modeling and simulation projects and studies underlying this paper. P.L. Hughes skillfully edited several versions of the manuscript.

REFERENCES

[1] http://www.iec.ch/functionalsafety/explained/ IEC 61508 Version 1.0 published in 2005 and Version 2.0 in 2010

[2] http://www.iso.org/iso/catalogue_detail?csnumber=43464 The Draft ISO 26262 Standard was published in 2009 and the 1st edition of the final standard in2011.

[3] http://www.autosar.org/index.php?p=1&up=1&uup=2&uuup=0

[4] M.D. Natale and A.L. Sangiovanni-Vincentelli, "Moving from Federated to Integrated Architectures for Automotive", Proc. IEEE, Vol. 98, No. 4, 603-620, April 2010.

[5] E.A Lee and H. Zheng, "Operational Semantics of Hybrid Systems", invited paper in Proc. of Hybrid Systems: Computation and Control (HSCC), LNCS 3414, Zurich, Switzerland, March 9-11, 2004.

[6] R.C. Conand and W.R. Ashby,"Every good regulator of a system must be a model of the system",Intl. Jrnl of Syst. Science, Intl. Journal of Systems Science", Vol. 1, pp 87-97, 1970.

[7] http://en.wikipedia.org/wiki/Vehicle_dynamics

[8] T.D. Gillespie, "Fundamentals of vehicle dynamics (2nd Ed.)". Warrendale, PA: Society of Automotive Engineers1992. ISBN 978-1-56091-199-9.

[9] Gipps, P.G. "Computer Program MULTSIM for Simulating Output from Vehicle Detectors on a Multi-Lane Signal Controlled Road", Transport Operations Research Group Working Paper 20, University of Newcastle-Upon-Tyne, 1976. http://en.wikipedia.org/wiki/Gipps%27_model

[10] www.carsim.com

[11] M.V. Stringfellow, N.G. Leveson and B.D Owens, "Safety-Driven Design for Software-Intensive Aerospace and Automotive Systems", Proc. IEEE, Vol. 98, No. 4, 515-525, April 2010.

[12] A. Abdallah, E.M. Feron, G.R. Hellestrand and M. Wolf, "Hardware/Software Codesign of Aerospace and Automotive Systems", Proc. of IEEE, Vol. 98, No.4, 584-602, Apr 2010.