Methods and Tools for Systems Engineering of Automotive Electronic Architectures

Jakob Axelsson
Volvo Technological Development Corp.
SE-412 88 Göteborg, SWEDEN
E-mail: jakob.axelsson@vtd.volvo.se

Abstract

The latest generations of road vehicles have seen a tremendous development in on-board electronic systems, which control increasingly large parts of the functionality. In this paper, we discuss how the vehicle manufacturers need to adjust their methods and tools to handle the increasing complexity. The key issue is the system integration aspect, which calls for increasing systems engineering capabilities.

1. Automotive electronics

The automotive industry is currently going through a dramatic increase of electronic equipment for on-board vehicle control. Until quite recently, the electrical system in a vehicle was mostly concerned with lights, starter motors, windshield wipers, and the like. In modern cars, tens of electronic control units (ECUs) implement sophisticated real-time control functionality. The ECUs are placed as closely as possible to the parts they control, and are connected to each other via distribution networks.

To handle the added complexity, the industry must find new development methods, and in particular the requirements capturing and system analysis must be adapted. The role of the automotive manufacturer is increasingly becoming that of a system integrator working closely with suppliers that take a large responsibility for the subsystems. This change in focus leads to a need for new ways of organizing the development work, and new ways of exchanging information about the product. The key capability is that of systems engineering, which ensures that a holistic perspective of the product is preserved, thereby guaranteeing those requirements that are of greatest importance to the users. Also, the whole life cycle of the product is taken into account, including production and after market requirements and processes.

The electronic system in a vehicle is closely connected to the vehicle itself, and the requirements on the complete vehicle are reflected in the electronic architecture. The mission of the vehicle is to transport passengers and goods efficiently. This means that factors such as cost, safety, weight, availability, environmental impact, comfort, and information handling are of key importance. The electronic architecture plays an important role in fulfilling all these requirements, through e.g. ABS brakes, engine management, climate control, and instrumentation.

There is a trend to replace or enhance traditional mechanical solutions with electronics. This includes safety systems, such as stability control, but also fundamental vehicle functionality such as brakes and steering, where electronics can provide increased functionality and at the same time reduce cost and weight. However, new electronic systems, such as fault-tolerant buses, are likely to be needed to reach sufficient safety levels for these crucial functions.

Another area where large changes are anticipated is in information to the driver and passengers. This includes both navigation systems, Internet access, fleet management for trucks and buses, entertainment for the passengers, etc., and will affect the technology used in the electronic systems since current networks have insufficient bandwidth. Most likely, fibre optical solutions will be required to reach speeds in the Mbit range. This kind of equipment also puts different requirements on the flexibility, since the owner might wish to upgrade the vehicle by putting in e.g. new audio or video equipment after the vehicle is delivered.

2. System integration

In the automotive industry, there is little doubt that the electronics growth will continue, and to remain competitive it is crucial to master the new technology. At the same time, the cost pressure does not allow the vehicle manufacturers to develop their own solutions to all functionality. There is a long tradition in the industry to work with suppliers, and a number of suppliers exist that deliver similar systems to all vehicle manufacturers. The role of the vehicle manufacturer is then to specify the subsystems to be purchased, and integrate them in the vehicle.

So far, the suppliers have mostly developed isolated subsystems, e.g. consisting of an ECU, its software, and possibly mechanical parts, too. However, in the future it is expected that the number of ECUs will not grow drastically, simply because it becomes too expensive and heavy to have separate hardware for each functionality, and the available space in the vehicle is also limited. Therefore, ways must be found to integrate software modules that participate in the implementation of different functionality in the same ECU. The software modules may be delivered by different suppliers, and the integration must ensure the integrity and performance of all the functions despite the sharing of hardware resources. This is a particular challenge when it comes to safety-critical functionality, and the requirements on the software modules are thus not limited to their functional interface, but touches their overall behaviour. In a sense, the automotive software modules are a kind of intellectual property (IP), similar to the behavioural IP modules that are used for designing systems on chip, and that can be traded between different system integrators. The differences lay mainly in the distributed nature of the automotive systems, and in their close integration with the vehicle itself.

To handle development which relies heavily on the use of IP, it is necessary to have a generic software architecture similar to the one in Figure 1 which determines how the IPs are interfacing to each other, and how resources are shared between them. The functional layer is likely to be specified by the vehicle manufacturer, since it defines the functionality of the vehicle. This specification is partitioned into software modules that are allocated to different ECUs. The software is developed by the suppliers, and the ECU hardware by, possibly, other suppliers. The vehicle manufacturer integrates the software modules, and to facilitate the integration, a standardized support layer is used, which hides the details of the hardware from the software modules, and defines interfaces for inter-module communication. The communication works equally well internally in an ECU and externally, over the network, and it is the role of the vehicle manufacturer to maintain the signal information. The hardware layer is mostly built from standard components.

3. Systems engineering methods & tools

To turn into efficient system integrators, it is clear that the vehicle manufacturers must become better at specifying the functionality of the system, and at making trade-offs in order to find the best system architecture to suit the needs expressed in the requirements. This challenge is met by

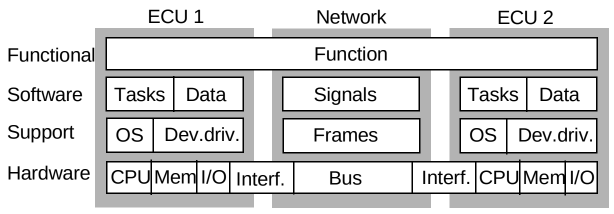

Figure 1. A layered view on the implementation of an automotive function.

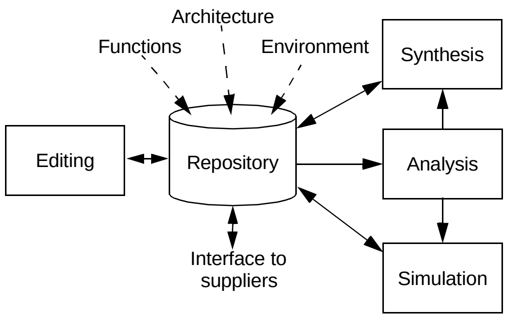

Figure 2. System-level development tools.

improving the systems engineering (SE) capability, since SE addresses exactly these issues. It also improves the perspective of the system life-cycle, and covers many aspects of the system simultaneously, since the electronics are so tightly integrated with the rest of the vehicle.

SE is very much an organizational issue, since it relies on the discipline and maturity of the engineers performing the work. However, once the organization is ready, tools may be introduced to support different activities. One of the crucial issues is to handle all the information related to the system, and in order to maintain a holistic view of the product and its supporting processes, a common repository in the form of a product data management system is necessary, to which different tools can be interfaced in order to assist the developers. Figure 2 shows the overall organization of the systems engineering tools.

It is our belief that the only realistic candidate for such a common information repository is object-oriented models, since they are based on natural concepts that are present in both hardware and software, as well as in other domains. Standards like UML are a reasonable basis, but new ways of using them must be found in order to cover the needs of systems engineering rather than software engineering, for which the notion was originally developed. This includes in particular quantitative analysis of e.g. the continuous behaviour of physical components, or the system cost.

Among the tools that are needed in the system-level activities are:

- **Simulation** to get an early functional validation.
- **Analysis** to estimate cost, resource utilization, response times, fault-tolerance, etc.
- **Synthesis** to automatically perform certain design activities, e.g. software code generation.
- **Requirements handling** to ensure that the requirements are consistent and that solutions are traceable.
- **Configuration management** to make sure that the repository remains consistent and up to date.

In the future, one can also imagine the inclusion of more advanced synthesis tools, that perform automatic partitioning or even automatic derivation of the overall system architecture. Since these are mainly optimization problems, good analysis models is the enabling factor to automation.