A proposal for an Automotive Architecture Framework for Volvo Cars

Patrizio Pelliccione†, Eric Knauss†, Rogardt Heldal†, Magnus Ågren†, Piergiuseppe Mallozzi†,
Anders Alminger*, Daniel Borgentun*
*Volvo Cars, Sweden - {anders.alminger,daniel.borgentun}@volvocars.com
†Chalmers University of Technology | University of Gothenburg
Department of Computer Science and Engineering, Sweden
{eric.knauss, patrizio.pelliccione}@gu.se, {heldal, magnus.agren, mallozzi}@chalmers.se

Abstract—**During the past twenty years vehicles have become more and more robot like, interpreting and exploiting input from various sensors to make decisions and finally commit actions that were previously made by humans. Such features will require continuous evolution and updates to ensure safety, security, and suitability for supporting drivers in an ever changing world. Modern vehicles can have over 100 Electronic Control Units (ECUs), which are small computers, together executing gigabytes of software. ECUs are connected to each other through several networks within the car, and in some cases also to the outside world. This need for addressing ever increasing complexity as well as for offering flexibility, support of continuous evolution, and very late changes in user visible features introduces new challenges for developing and maintaining a suitable electronic architecture. In this paper we report the current investigation of the Volvo Cars to create an architecture framework tailored to the needs of future vehicles.**

I. Introduction

Historically, software was introduced in cars to optimise the control of the engines. Since then the growth of software within the car has been exponential for each year and today not a single function is performed without the involvement of software. According to industry experts, 80% to 90% of the innovation within the automotive industry is based on electronics. A big part of electronics is software.

This evolution of the automotive industry creates new challenges regarding software architecture development and maintenance. The increasing amount of people involved in the software development projects imposes additional challenges to the architecture teams, as the development and design literally cannot be controlled, or even understood, in detail by a single group any more. The development is inevitably parallelized; this obviously also holds for the large amounts of externally developed software, which is integrated as black box functionality. The important integration work is done in an iterative manner by developers and test teams, focusing on vital systems first and then gradually establishing various functionalities. Architects might be not involved in integration, however, the architecture for sure influences the integration.

In this paper we report a current initiative of the Volvo Cars to renovate the electrical architecture. The work is part of a Vinnova FFI Swedish project, called Next Generation Electrical Architecture (NGEA)[1], which mainly focuses on the following areas: (i) continuous integration and deployment; (ii) cars as constituents of a system of systems; (iii) reducing the number of ECUs through an architecture that allows the identification of key functions to be implemented in domain nodes; and (iv) strategies to improve the automotive ecosystem so to enable rapid communication with suppliers and flexible development.

Within the project we are investigating the possibility to create a Volvo Cars Architecture Framework [11]. We believe that an architecture framework, together with its multiple viewpoints, is the instrument to manage the increasing complexity of modern vehicles. It aims at ensuring that descriptions of vehicle architectures can be compared and related across different vehicle programs, development units and organizations, thus increasing flexibility and innovation, while reducing development time and risks.

We build on existing architecture frameworks in the automotive domain, i.e. [2], [4] and we base our work on the conceptual foundations provided by the ISO/IEC/IEEE 42010:2011 standard [11]. The standard addresses architecture description, i.e. the practices of recording software, system and enterprise architectures so that architectures can be understood, documented, analysed and realized.

The paper is organized as follows. Section II gives an overview of the state of the art on architecture framework in the automotive domain. Section III analyses the state of practice in the context of Volvo Cars and reports some of the lessons learned. Section IV describes the first steps towards the definition of an architecture framework for Volvo Cars. Finally the paper concludes in Section V with final remarks and a discussion about future work.

II. Automotive Architecture Frameworks

An architecture framework is a coordinated set of viewpoints, conventions, principles and practices for architecture description within a specific domain of application or community of stakeholders [11]. More specifically, it is determined by: (i) a set of architecture-related concerns, (i) a set of

[1]http://www.vinnova.se/sv/Resultat/Projekt/Effekta/2009-02186/Next-generation-electrical-architecture/

978-1-5090-2571-8/16 $31.00 © 2016 IEEE
DOI 10.1109/WASA.2016.9

stakeholders holding those concerns, (ii) a set of architecture viewpoints which frame (i.e., cover) those concerns, and (iv) a set of model correspondence rules to impose constraints between types of models and then demonstrate that constraints are satisfied by the architecture.

An architecture framework is a prefabricated knowledge structure, identified by *architecture viewpoints*, that architects use to organize an architecture description into *architecture views*. The terms architecture view and architecture viewpoint are central to the standard [11]. An *architecture viewpoint* encapsulates notations, conventions, methods and techniques to be used according to specific concerns and for a particular audience of system stakeholders.

Many existing practices express architectures through collections of models, and models are further organized into cohesive groups, called *views*. A view can be defined as a "*work product expressing the architecture of a system from the perspective of specific system concerns*" [11].

For further discussion of the content model and architecture frameworks mechanism, see [6]. For an extensive survey of frameworks, see [10].

Architecture frameworks have not been standardized in the automotive domain and automotive industry. However, some attempts exist and different types of architecture viewpoints and views have been introduced recently as part of automotive architecture frameworks.

1) Automotive Architecture Framework: A first attempt towards a standardized architectural foundation and automotive-specific architecture framework is the Automotive Architecture Framework (AAF) proposed in [2]. The purpose of AAF is to describe the entire vehicle system across all functional and engineering domains and drive the thought process within the automotive industry. The AAF conforms to the ISO 42010 international standard [11], and therefore it is defined in terms of a set of viewpoints and views.

The AAF distinguishes between two sets of architecture viewpoints and views: (i) mandatory and general viewpoints and (ii) optional viewpoints. The following viewpoints are presented according to the viewpoint catalog in [19]. The mandatory viewpoints and their respective views include: (i) Functional viewpoint - functional decomposition, functional architecture; (ii) Logical viewpoint - logical decomposition, logical architecture; this viewpoint is only mentioned in [2] but not detailed in [1]; (iii) Technical viewpoint - physical decomposition, technical architecture; (iv) Information viewpoint - perspective of information or data objects used to define and manage a vehicle; (v) Driver/vehicle operations viewpoint - vehicle environment; (vi) Value net viewpoint - OEM stakeholders and those of its suppliers and engineering partners. The optional viewpoints suggested by the AAF are: (i) Safety, (ii) Security, (iii) Quality and RAS - Reliability, Availability, Serviceability, (iv) Energy, possibly including performance, (v) Cost, (vi) NVH - noise, vibration, harshness, and (vii) Weight.

2) Architectural Design Framework: The Architectural Design Framework (ADF) [7] is developed by Renault and includes operational, functional, constructional, and requirements viewpoints. Although the AAF and ADF are related they have some differences.

- *Operational Viewpoint* - this is the more abstract viewpoint. The system is observed from a black box and user perspective [7].
- *Functional Viewpoint* - system functions identified in the views associated to the operational viewpoint are allocated to SysML Blocks.
- *Constructional Viewpoint* - this viewpoint describes a further allocation (or grouping) of system functions into physical components.
- *Requirements Viewpoint* - This viewpoint is orthogonal to other viewpoints. Each requirement view has a relationship with views of other viewpoints.

3) Architecture Framework For Automotive Systems: The Architecture Framework for Automotive Systems (AFAS) is proposed in [4] through an analysis of AAF, ADF and of Architecture Description Languages [17], [16] tailored to automotive domain, like EAST-ADL [3] and AML [15]. It contains an elaboration and unification of the viewpoints proposed in AAF and ADF and then proposes additional viewpoints, e.g.:

- *Feature viewpoint* - to be used to support the product line engineering.
- *Implementation viewpoint* - it describes the software architecture of the Electrical/Electronic (E/E) system in the vehicle [3].

III. Lessons learnt within Volvo Cars

In a previous paper we made a study within Volvo Cars to identify, from the architecture point of view, the challenges that OEMs are facing in the last years [5].

We identified that there is not always an obvious connection between architecture (or top-level design) and design, especially during later development. The architecture is only communicated as large documents that are supposed to be read by stakeholders. However, this not always corresponds to the reality. Maintenance of the architecture, while the design evolves, is demanding and not always performed in all parts. This shows a discrepancy between the planned architecture defined according to a V-Model process, and the architecture that is actually emerging from the system development.

On the organizational side, we found a need to improve the communication between different groups, for instance by making teams more cross-functional. Espousing the terminology in [14], system architects should be also "Extrovert" architects (conceptually related to the external focus of [13]), i.e. devoted to communicating the architectural decisions and knowledge to the other stakeholders.

It emerges the need to explore both organizational and technical possibilities for tighter cooperation between architecture levels, and to measure effects such improvements would have. Solutions are needed to support agile architecting [22] as well as to enable stakeholders different from the architects, such as developers, to improve the architecture, such as fixing wrong assumptions or making decision deliberately postponed.

Another interesting finding is that the architecture is not clearly considering the highly complex supplier-network that characterizes automotive engineering. In a previous study [23] we found that a holistic strategy for aligning work across the value-chain is currently missing. Specifically, mixing commodity and differentiating components lead to sub-optimal situations, resulting in duplicated work. We argue that automotive architecture needs to assume a holistic perspective with respect to the whole value-chain and optimize the architecture for facilitating beneficial subcontractor interaction. This calls for a proper management of the automotive ecosystem, which is characterized by relying heavily on complex supplier networks, and strong dependence on hardware and software development [12].

IV. Volvo Cars architecture framework

The starting point for defining an architecture framework is to start from the identification of established stakeholders within the domain of the framework. Stakeholders may be individuals, teams, organizations or classes (of individuals, teams or organizations), while concerns may be fine-grained or very broad in scope [6]. Then the identified stakeholders motivate the set of concerns on which the architecture framework will focus. This will help the consumer of the architecture framework and of the views and connected modeling tools to understand why they are modeling and when they are done.

The identified architecture-related concerns determine the choice of viewpoints and view to be included. It is important to note that almost each viewpoint detained in the following is already handled within the current architecture of Volvo Cars. However, we are investigating the definition of proper viewpoints that will create the architecture framework. Therefore, in addition to the viewpoints summarized in Section II we foresee the following viewpoints:

- *Continuous integration and deployment* (detailed in Section IV-A) - OEMs are increasingly interested to reduce the development time, to increase flexibility, to have early feedback on decisions made, and to add new functionalities incrementally even after production.
- *Connected cars and safety* (detailed in Section IV-B) - Future scenarios of collaborating autonomous vehicles, like platooning, will require to extend the vehicle safety architecture across the classical boundaries of single vehicles and will ask for an open and adaptive architecture able to support runtime assessment of safety.
- *Security and privacy of connected cars* - Connected cars open new important challenges from the point of view of security and privacy.
- *Ecosystem and transparency* - related to the value net viewpoint of AAF [1], [2]. The ecosystem around the OEM can be seen as a virtual organization consisting of the OEM, its suppliers and other partners involved in the process of creating customer value.
- *Autonomous cars* - autonomous cars require special architecture solutions, e.g. inspired to autonomous and self-adaptive systems [20].
- *Modes management* - a mode viewpoint is needed to design the different modes of a vehicle as well as the transitions from one mode to another.
- Special viewpoints and views might be conceived to enable dissemination and communication of the architecture to developers and other stakeholders.

The natural consequence of the use of multiple viewpoints and views in architecture descriptions is the need to express correspondences and consistency rules between those views. The mechanisms introduced in [11] is called model correspondences and it allows the definition of relations between two or more architecture models. Since architecture views are composed of architecture models [11], model correspondences can be used to relate views to express consistency, traceability, refinement or other dependencies [6]. These mechanisms allow an architect to impose constraints between types of models and then demonstrate that them are satisfied by the architecture.

A. Continuous integration and deployment

Agile approaches and practices such as continuous integration and deployment promise to help reducing development time, to increase flexibility, and to generally shorten the feedback cycle time. However, the complex supplier network, and typical setup with a large number of ECUs, pose specific challenges to these practices.

First, dependencies between ECUs raise multiple concerns, regarding organization, versioning and testing: (i) organization - identifying the recipient of a given software change; (ii) versioning - the question is related to the compatibility of the software version of specific ECUs; and (iii) testing - compatible combinations need to be validated. Second, support for continuous deployment has to face with safety concerns. Should, for instance, the software of an ECU responsible for a safety critical function be modifiable at runtime?

Dependencies between ECUs are a property of the architecture. As mentioned, the emergent architecture may differ from the intended architecture, and continuous integration and deployment of software may entail architectural changes. This highlights both the need for collaboration between parts of the organization working on different architectural levels, and the need of a proper support for agile and flexible architecting.

Addressing these concerns suggests two architectural views and viewpoints: (i) one covering architecture as an enabler of continuous integration and deployment, facilitating variant handling and coordination of updates, and (ii) another considering continuous integration and deployment on the architecture level, facilitating reasoning about modifications to the architecture itself.

B. Connected cars and Safety

Connected cars will benefit from Intelligent Transport Systems (ITS), Smart Cities and IoT to provide new application scenarios like smart traffic control, smart platooning coordination, collective collision avoidance, etc. Vehicles will combine data collected through its sensors with external data coming from the environment, e.g. other vehicles, road, cloud, etc.

Connected vehicles will face new challenges and opportunities related to safety issues. Current regulations for safety aspects, like the ISO 26262 standard, do not account for scenarios in which the vehicle is part of a more complex system; the challenge is on how to manage new hazards that coming from the environment could jeopardize safety. At the same time, connected cars open new opportunities for safety, called "connected safety" within Volvo Cars[2]: e.g., this new technology will allow a connected car to be aware of a slippery road, of cyclists on the road, etc., so to initiate all the actions needed to avoid accidents and collisions.

These scenarios are posing new requirements to the architecture. We foreseen two different viewpoints and views: (i) from the point of view of a single connected car, which has to offer the functionalities needed to realize the scenario, and (ii) from the point of view of the system of systems (i.e. cars connected with other entities of their environment), spanning from the agreements between the different systems affected, e.g. different OEMs, cloud providers, road infrastructures, etc., to the definition of the scenario that the system of systems has to realize, like the slippery road mentioned above.

In general, the main issue with safety guarantees in connected cars is that a full analysis of the system is not possible at design time. When moving from a single vehicle to a cooperative system, a new safety analysis is required to handle uncertainties at runtime. Some approaches have been proposed to deal with certification at runtime, e.g. [21], [18], but a clear framework that can be used to define the connected safety requirements is still missing.

V. Discussion and Concluding remarks

In this paper we described the current evaluation of Volvo Cars towards the definition of an architecture framework. In this stage we are identifying potential viewpoints of the vehicles of the near future. We will then identify the modeling languages to be used to describe the architecture. According to the actual needs and to the consumers of the particular view, we will identify the modeling language to be used and we will find the right tradeoff between a language (i) able to support the level of formality and precision required by disciplined development processes, or (ii) simple and intuitive enough to communicate the right message to stakeholders and to promote collaboration [16], [14].

For what concerns the realization of the architecture framework we will investigate MEGAF [8], [9], which is an infrastructure that enables software architects to realize their own architecture frameworks specialized for a particular application domain or community of stakeholders and compliant to the ISO/IEC/IEEE 42010 standard [11].

Acknowledgment

This work was partially supported by the NGEA Vinnova project and by the Wallenberg Autonomous Systems Program (WASP).

[2]https://goo.gl/mIWWS3

References

[1] M. Broy, M. Gleirscher, P. Kluge, W. Krenzer, S. Merenda, and D. Wild. Automotive Architecture Framework: Towards a Holistic and Standardised System Architecture Description. Technical report, 2009.

[2] M. Broy, M. Gleirscher, S. Merenda, D. Wild, P. Kluge, and W. Krenzer. Toward a holistic and standardized automotive architecture description. *Computer*, 42(12):98–101, 2009.

[3] P. Cuenot, P. Frey, R. Johansson, H. Lönn, Y. Papadopoulos, M.-O. Reiser, A. Sandberg, D. Servat, R. T. Kolagari, M. Törngren, and M. Weber. The east-adl architecture description language for automotive embedded software. In *Proceedings of MBEERTS'07*, pages 297–307. Springer-Verlag, 2010.

[4] Y. Dajsuren. On the design of an architecture framework and quality evaluation for automotive software systems, PhD thesis. Technische Universiteit Eindhoven, 2015.

[5] U. Eliasson, R. Heldal, P. Pelliccione, and J. Lantz. Architecting in the automotive domain: Descriptive vs prescriptive architecture. In *Proceedings of WICSA2015*, pages 115–118, May 2015.

[6] D. Emery and R. Hilliard. Every architecture description needs a framework: Expressing architecture frameworks using ISO/IEC 42010. In *WICSA/ECSA 2009*, 2009.

[7] H. G. C. Góngora, T. Gaudré, and S. Tucci-Piergiovanni. Towards an architectural design framework for automotive systems development. In *Complex Systems Design & Management*, pages 241–258. Springer Berlin Heidelberg, 2013.

[8] R. Hilliard, I. Malavolta, H. Muccini, and P. Pelliccione. Realizing architecture frameworks through megamodelling techniques. In *Proceedings of ASE '10*, pages 305–308. ACM, 2010.

[9] R. Hilliard, I. Malavolta, H. Muccini, and P. Pelliccione. On the composition and reuse of viewpoints across architecture frameworks. In *Proceedings of WICSA-ECSA '12*, pages 131–140. IEEE Computer Society, 2012.

[10] ISO/IEC-JTC1/SC7-WG42. Survey of architecture frameworks.

[11] ISO/IEC/IEEE. *ISO/IEC/IEEE 42010, Systems and software engineering — Architecture description*, December 2011.

[12] E. Knauss and D. Damian. Towards enabling cross-organizational modeling in automotive ecosystems. In *Proceedings of MD2P2*, pages 38–47, 2014.

[13] P. Kruchten. What do software architects really do? *Journal of Systems and Software*, 81(12):2413 – 2416, 2008.

[14] P. Lago, I. Malavolta, H. Muccini, P. Pelliccione, and A. Tang. The road ahead for architectural languages. *Software, IEEE*, 32(1):98–105, Jan 2015.

[15] M. V. M. Rappl, P. Braun and C. Schröder. Automotive Software Development: A Model Based Approach. In *World Congress of Automotive Engineers, SAE Technical Paper Series 2002-01-0875*, 2002.

[16] I. Malavolta, P. Lago, H. Muccini, P. Pelliccione, and A. Tang. What industry needs from architectural languages: A survey. *IEEE Transactions on Software Engineering*, 39(6):25, June 2013.

[17] N. Medvidovic and R. N. Taylor. A Classification and Comparison Framework for Software Architecture Description Languages. *IEEE Transactions on Software Engineering*, 26(1), January 2000.

[18] K. Östberg and M. Bengtsson. Run time safety analysis for automotive systems in an open and adaptive environment. In *Proceedings of SAFECOMP 2013-Workshop ASCoMS*, 2013.

[19] N. Rozanski and E. Woods. *Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives*. Addison-Wesley Professional, 2005.

[20] M. Salehie and L. Tahvildari. Self-adaptive software: Landscape and research challenges. *ACM Trans. Auton. Adapt. Syst.*, 4(2):14:1–14:42, May 2009.

[21] D. Schneider and M. Trapp. Conditional safety certification of open adaptive systems. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 8(2):8, 2013.

[22] A. Shahrokni, P. Gergely, J. Söderberg, and P. Pelliccione. Organic Evolution of Development Organizations-An Experience Report. SAE Technical Paper, 2016-01-0028, 2016, doi:10.4271/2016-01-0028.

[23] M. Soltani and E. Knauss. Challenges of requirements engineering in autosar ecosystems. In *Proceedings of 23rd IEEE International Requirements Engineering Conference*, Ottawa, Canada, 2015.