Automotive Architecture Description and Its Quality

Yanja Dajsuren
TU/e campus MF 7.92
Eindhoven University of Technology
5612 AZ Eindhoven, The Netherlands
y.dajsuren@tue.nl

ABSTRACT

This research is part of the Hybrid Innovations for Trucks (HIT), an ongoing multi-disciplinary project with the objectives of CO_2 emission reduction and fuel saving for long haul vehicles. Achieving this goal necessitates definition of a proper architecture and quality techniques to enable the development of a new and more efficient control software. Therefore, this research covers automotive architecture description language and quality of automotive software.

Categories and Subject Descriptors

D.2.11 [**Software Engineering**]: Software Architectures—*Domain-specific architectures, Languages (* e.g., *description, interconnection, definition)*; D.2.8 [**Software Engineering**]: Metrics

General Terms

Automotive, Architecture, Design, Modeling, Languages, Measurement

Keywords

automotive architecture, architecture description language, architectural quality, quality metrics

1. INTRODUCTION

The Hybrid Innovations for Trucks (HIT) project is financially supported by the Dutch automotive innovation programme. The HIT project aims at increasing efficiency of the powertrain by developing a new and more efficient energy manager, a software component determining the optimal use of the available power resources to reduce the total fuel energy over a drive cycle [20]. The fact that energy management, functionality so crucial for the modern trucks, is delegated to software is indicative of the immanence of software in the automotive world. Indeed, since the introduction of software in vehicles thirty years ago, the amount of software has grown exponentially and nowadays is responsible for 50-70% of the total electronics and software development costs [4]. Hence, one of the project work packages focuses on identifying appropriate ways of developing automotive software, in general, and energy manager, in particular.

ESEC/FSE '13, August 18–26, 2013, Saint Petersburg, Russia

Automotive software engineering also known as software engineering for automotive systems cover almost all areas of computer science and computer engineering [21, 19, 14]. Enabling interaction between different engineering disciplines such as mechanical engineering, electrical engineering, and software engineering is considered a key prerequisite of automotive software development [21]. Therefore, an Architecture Description Language (ADL) is considered a viable solution to manage multi-disciplinary engineering information (*e.g.,* structure, behavior, requirements, variability, and verification and validation [21]) in an effective way. Recognising an importance of an ADL, automotive companies have been actively involved in the development of automotive ADLs *e.g.,* BMW in the definition of AML, Volvo, Fiat, and VW/Carmeq in the EAST-ADL and TADL. The EAST-ADL2 is being extended to model the fully electric vehicle in the scope of the ICT MAENAD project, where many automotive manufacturers and suppliers are participating. Besides the domain-specific modeling languages, SysML and MARTE are attracting considerable attention of automotive companies as well.

Hence, the following work package deliverables have been defined in the scope of this research:
1. Identify an existing or design a new automotive ADL, supporting requirements traceability, multi-level modeling, modeling hierarchical elements and evolution of models, means for determining the architectural quality of models, and the tool support for the aforementioned tasks.

To elicit the requirements that the modeling approach should satisfy, we conducted a series of structured interviews with five architects responsible for modeling automotive software at different architectural views, ranging from functional architecture to hardware architecture as illustrated in Figure 1. Since automotive ADLs do not explicitly address the quality of automotive system and software architectures [8], another work package task is dedicated to the study of quality of automotive system and software architecture:
2. Identify quality concerns relevant in the automotive architecture, and propose means of evaluating these concerns.

This paper focuses on the results obtained in this work package so far and outlines the directions that we will pursue till 2014. The remainder of this paper is organized as follows. In Section 2, a work related to the evaluation of automotive ADLs is summarised. In Section 3, enforcing architectural inconsistency between different automotive architectural views and defining quality metrics for architectural and design models are presented. In Section 4, conclusion and future works are discussed.

2. AUTOMOTIVE ADL AND AUTOSAR

An Architecture Description Language (ADL) is one of the approaches to formalize the representation of automotive system and software architecture. An ADL is defined as any form of expression for use in architecture descriptions [13]. Based on the requirements of automotive architectural modeling which are elicited during the interviews with the automotive architects, we have conducted a comparative study of existing ADLs [8] aiming at choosing the most appropriate ADL for the task. We have compared four automotive ADLs (EAST-ADL [7], AADL [10], TADL [23] and AML [3]) as well as more general system ADLs (SysML [17] and MARTE [18]). While each one of the languages studied had its advantages and disadvantages, we decided to focus on SysML due to its support for main requirements such as requirements traceability, different type of hierarchical elements, availability and accessibility of the specification as well as mature open-source and commercial tool support.

As a next step, we plan to investigate a relation between automotive ADL and AUTOSAR (AUTomotive Open System ARchitecture)[1], which is an open and standardized automotive software architecture developed by the automotive companies (OEMs), suppliers and tool developers.

3. QUALITY OF AUTOMOTIVE SYSTEM AND SOFTWARE ARCHITECTURE

In the evaluation of automotive ADLs [8], we identified that the automotive ADLs lack the capability of ensuring the architectural quality. We focus on architectural inconsistency and other quality attributes due to the lack of consistency enforcement of multiple architectural views and lack of quality (*e.g.,* modularity) definition among these views. Therefore, in Section 3.1, we discuss the preliminary results of the architectural inconsistency between the different architectural views as one of the key internal quality issues of automotive systems. In Section 3.2, we address the challenges and results of addressing quality attributes of automotive system and software architecture, specifically modularity metrics of Simulink models.

3.1 Architectural Inconsistency

According to the ISO42010 international standard for systems and software engineering [13], an ADL provides one or more model kinds (data flow diagrams, class diagrams, state diagrams etc.) as a means to frame some concerns for its stakeholders. An ADL can be widely focused to provide model kinds optionally organized into views.

As illustrated in Figure 1, we derived the following views based on proprietary automotive architectural models, the automotive architecture framework [5], and automotive ADLs, which together describe an entire vehicle system across all functional and electrical/electronical (E/E) domains [14]:

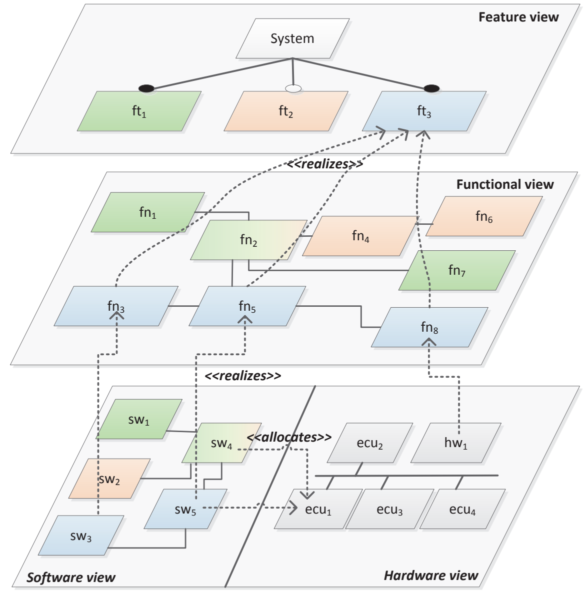

Figure 1: Automotive architectural views.

- *Feature view* captures the vehicle product line features, such as cruise control or bluetooth telephone connection, which can be configured for a product or a specific vehicle.
- *Functional view* contains a structural model kind that contains a number of functions or subsystems realizing features.
- *Software view* represents software architecture, where detailed descriptions and implementation of a function is realized in software components or blocks.
- *Hardware view* represents E/E hardware architecture. The hardware architecture typically consists of electronic control units (ECUs), sensors, actuators and CAN busses.

We proposed consistency rules as correspondence rules between automotive architectural views based on ISO42010 standard [11]. We focus on the realization relationship between functional and software views, where the functional models are refined by adding more details in the software view. A prototype tool was developed for IBM Rational Rhapsody which can perform this consistency checking between functional and software views. As a next step, we plan to carry out a comprehensive industrial case study with automotive architects.

3.2 Quality Metrics

Figure 2 illustrates an example representation of functional and software views of Figure 1 as a functional architecture in SysML and software architecture in Simulink. Automotive software is commonly being developed using model-based design tools like Simulink and Stateflow[2] together with automatic code generation tools. Assessing quality of MATLAB/Simulink/Stateflow models has become more important for automotive manufacturers due to the increasing complexity of the models and stricter safety-related requirements [12]. Large automotive Simulink models can consist of up to

[1] `http://autosar.org/`

[2] `http://www.mathworks.com/`

15,000 building blocks, 700 subsystems and 16 hierarchical levels [22]. Current quality assessment techniques such as the Mathworks Automotive Advisory Board (MAAB) guidelines and Model Advisor from Mathworks focus mainly on configuration settings and guideline conformance rather than model quality [12]. Although there are plethora of source code quality analysis tools available, methods for assessing quality of automotive architecture and design models are still limited. Therefore, we aim to define quality attributes for

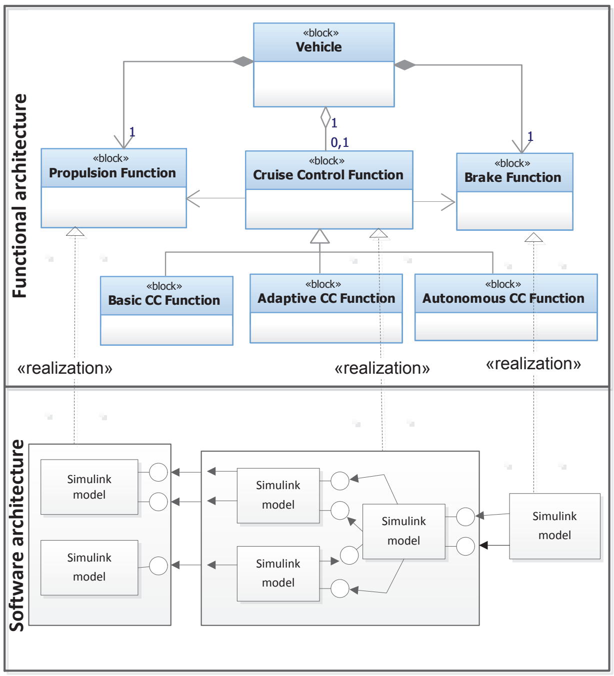

Figure 2: Illustrative example of automotive architectural models.

design models *i.e.,* Simulink models and aggregate them to the automotive system/functional and software architecture. We defined a quality model based on the ISO/IEC SQuaRe quality standard [1]. Our initial case study covered modularity metrics for Simulink models [9], since for automotive software modularity is recognized as being paramount [19] as changing or reusing non-modular software is very costly. Modularity is defined as a degree to which a system or computer program is composed of discrete components such that a change to one component has minimal impact on other components [1].

A quality model based on ISO/IEC 9126 standard for assessing internal quality of Simulink models is introduced by W. Hu *et al.* [12]. Six quality sub-characteristics like analysability, changeability, stability, testability, understandability, and adaptability are selected for the quality model together with respective metrics. However, modularity sub-characteristic and respective metrics are not explicitly addressed by this quality model and furthermore the metrics are not validated. Quality analysis metrics for measuring instability, abstractness, and complexity of Simulink models are introduced by Olszewska (Pląska) in [16] [15]. However, the modularity concerns are not explicitly addressed by Olszewska's metrics and the MathWorks tools.

Following the Goal-Question-Metrics (GQM) approach [2], we defined a modularity metrics suit consisting of 9 coupling and cohesion-related metrics and validated it with experts' evaluation. We identified three independent metrics based on the statistical analysis and identified a relation between modularity metrics and presence of errors. We have observed that high coupling metric values and high number of hierarchal levels (subsystem depth) frequently correspond to presence of errors. We developed a tool to measure modularity of the Simulink models and visualized the quality aspects with SQuAVisiT toolset [24]. We are planning to identify other insightful visualization for depicting metrics values to facilitate the maintenance effort.

We are investigating quality model extending modularity attribute and defining useful visualization of metrics and metrics evolution (measurement of different versions of software). Modularity is related to other quality characteristics *e.g., reusability*, *modifiability*, and *stability*. In the expert evaluation, understandability is considered as one of the key related quality characteristic as well due to the Simulink visual modeling. Therefore, we have extended the proposed modularity metrics with the related quality (sub-)characteristics for Simulink models and plan to aggregate them to the architectural level. Once the quality metrics for the architectural and design models for the automotive domain is thoroughly validated, it can be modified or extended further for other embedded domains.

4. DISCUSSION

In the automotive software work package of the HIT multidisciplinary project, automotive architecture and its quality are considered main issues given the increased use of software makes cars more difficult to repair due to its complexity [6]. Instead of defining yet another architecture description language, we investigated existing automotive ADLs and identified a generic ADL as a suitable language for representing an automotive architecture. Industrial applicability of the chosen ADL is still being continued in a pilot project to make the final selection. As a next step, a relation between automotive ADL and AUTOSAR will be investigated.

Identifying quality concerns relevant in the automotive architecture and defining an approach for improving automotive system and software architecture are another main concern of this research. Rhapsody plugin to detect automotive architectural inconsistency is developed and plan to be extended based on the comprehensive industrial case study with automotive architects. MATLAB/Simulink is one of the most popular graphical modeling languages and a simulation tool for validating and testing control software systems. Due to the increasing complexity and size of Simulink models of automotive software systems, it has become a necessity to effectively maintain the Simulink models. Furthermore, the subsystems or control functions at system/functional architecture are realized by the Simulink models, which remain as a black-box component. Therefore, measuring quality attributes of Simulink models and subsystems enable the possibility to define the quality of automotive architecture. We defined modularity metrics and carried out preliminary evaluation of the metrics. This will be extended by other quality attributes like reusability, modifiability, and understandability. We are investigating useful visualization of metrics and metrics evolution using the SQuAVisiT toolset.

5. ACKNOWLEDGEMENT

I would like to thank the Dutch automotive innovation programme HTAS (High-Tech Automotive Systems) of Ministry of Economic Affairs, Agriculture and Innovation, the Netherlands for funding the Hybrid Innovations for Trucks (HIT) (HTASI10002) project, and my advisors Prof. Dr. Mark. G. J. van den Brand and Dr. Alexander Serebrenik, Eindhoven University of Technology, the Netherlands.

6. REFERENCES

[1] ISO/IEC 25010:2011 systems and software engineering - systems and software quality requirements and evaluation (SQuaRE).

[2] V. R. Basili. Software modeling and measurement: the goal/question/metric paradigm. Technical report, College Park, MD, USA, 1992.

[3] P. Braun and M. Rappl. A model-based approach for automotive software development. In *OMER*, pages 100–105, 2001.

[4] M. Broy. Challenges in automotive software engineering. In *Proceedings of the 28th international conference on Software engineering*, ICSE '06, pages 33–42, New York, NY, USA, 2006. ACM.

[5] 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, dec. 2009.

[6] R. N. Charette. This car runs on code. `http://www.spectrum.ieee.org/feb09/7649`, 2009.

[7] 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. Springer Verlag, 2010.

[8] Y. Dajsuren, M. G. J. van den Brand, A. Serebrenik, and R. Huisman. Automotive ADLS: a study on enforcing consistency through multiple architectural levels. In *Proc. of the 8th international ACM SIGSOFT conference on Quality of Software Architectures*, pages 71–80. ACM, 2012.

[9] Y. Dajsuren, M. G. J. van den Brand, A. Serebrenik, and S. Roubtsov. Simulink models are also software: Modularity assessment. In *Proc. of the 9th international ACM SIGSOFT conference on Quality of Software Architectures*, 2013 (to appear).

[10] P. H. Feiler, D. P. Gluch, and J. J. Hudak. The architecture analysis & design language (AADL): An introduction. Technical Report CMU/SEI-2006-TN-011, Software Engineering Institute, Carnegie Mellon University, 2006.

[11] C. M. Gerpheide and Y. Dajsuren. Developing a consistency-checking tool for automotive architectural models. In *SET seminar report*, 2012.

[12] W. Hu, T. Loeffler, and J. Wegener. Quality model based on ISO/IEC 9126 for internal quality of MATLAB/Simulink/Stateflow models. pages 325 –330, march 2012.

[13] ISO/IEC/IEEE 42010:2011. Systems and software engineering - architecture description. http://www.iso-architecture.org/ieee-1471, 2011.

[14] N. Navet and F. Simonot-Lion. *Automotive Embedded Systems Handbook*. Industrial Information Technology. Taylor & Francis, 2008.

[15] M. Olszewska (Pląska). Simulink-specific design quality metrics. Technical report, Turku, Finland, 2011.

[16] M. Olszewska (Pląska), M. Huova, M. Waldén, K. Sere, and M. Linjama. Quality analysis of simulink models. In I. Schieferdecker and S. Goericke, editors, *Software Quality Engineering. Proceedings of the CONQUEST 2009.*, pages 223–240. dpunkt.verlag GmbH, 2009.

[17] OMG. Systems Modeling Language (SysML) Specification version 1.2. `http://www.sysml.org/specs`, 2010.

[18] OMG. UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded Systems Version 1.1. `http://www.omg.org/spec/MARTE/1.1`, 2011.

[19] A. Pretschner, M. Broy, I. H. Kruger, and T. Stauner. Software engineering for automotive systems: A roadmap. In *Future of Software Engineering*, pages 55–71, Washington, DC, USA, 2007. IEEE Computer Society.

[20] V. van Reeven, T. Hofman, R. Huisman, and M. Steinbuch. Extending energy management in hybrid electric vehicles with explicit control of gear shifting and start-stop. In *American Control Conference (ACC), 2012*, pages 521–526, june 2012.

[21] J. Schäuffele and T. Zurawka. *Automotive Software Engineering: Principles, Processes, Methods, And Tools*. R: Society of Automotive Engineers. SAE International, 2005.

[22] I. Stürmer and H. Pohlheim. Model quality assessment in practice: How to measure and assess the quality of software models during the embedded software development process. In *Embedded Real Time Software and Systems*, 2012.

[23] The TIMMO Consortium. TADL: Timing Augmented Description Language version 2. `http://www.timmo-2-use.org`, 2009.

[24] M. G. J. van den Brand, S. Roubtsov, and A. Serebrenik. SQuAVisiT: A flexible tool for visual software analytics. In *Proceedings of the European Conference on Software Maintenance and Reengineering*, pages 331–332, Washington, DC, USA, 2009. IEEE Computer Society.