Boosting the Development of High-performance Automotive Systems

Alessio Bucaioni[1]

School of Innovation, Design and Engineering
Mälardalen University, Sweden
{name.surname}@mdh.se

Abstract. This paper describes a postdoctoral research aiming at improving the design of high-performance automotive systems through automation by model transformations. The main goal of this research is to enable early timing verification and feedback loops between verification and design of high-performance automotive systems. The research described in this paper is framed within the Swedish research project Automation in High-performance Cyber Physical Systems Development. More information about the project can be found at the following link: http://www.es.mdh.se/projects/520-Automation_in_High_performance_Cyber_Physical_Systems_Development

Keywords: Model-based software development, automotive systems, heterogeneous platforms, timing verification.

1 Introduction

The terrific data-throughput induced by contemporary vehicle functions is bringing automotive systems in the era of high-performance computing (HPC) [21]. In the automotive domain, HPC is realised by employing heterogeneous hardware platforms being a combination of certified real-time micro-controller units and general purpose HPC processors and accelerators (e.g., graphics processing units (GPUs), field-programmable gate arrays (FPGAs), etc.) [14]. However, when several computational units are put together on a single board, a crucial challenge arises that is how to use the enormous computing capabilities, while still meeting the system timing requirements. Timing verification is crucial for the development and safety certification of automotive systems, and real-time systems in general, as it provides evidences that their output will be delivered at the time that is suitable for the environment they interact with [19]. Schedulability analysis is a priori timing analysis technique which provides evidence on whether each function in the system is going to meet its timing requirements [19]. However, schedulability analysis for high-performance automotive systems is complicated by peculiar issues of heterogeneity such as allocation of computations to computational units. Allocation of computations to computational units might be an extremely tedious task as its optimisation often involves investigating the whole space of possible allocations, which is typical infeasible without automation support. Nowadays, there is little to no automation support in the development of high-performance automotive systems and allocation of computations to computational units is mostly done manually, which makes it tedious, error-prone and inefficient. One way to introduce automation in the development of high-performance automotive systems is by means of model-based techniques that is (meta)modelling and model transformations [17]. Models can be employed for representing the automotive software architecture and its timing-related properties while abstracting away from platform-specificity [7] Model transformations can provide automation support to schedulability analysis by automatically generating the design space of possible allocations [18].

In this paper, we describe a postdoctoral research aiming at improving the design of high-performance automotive systems through automation by model transformations. The main goal of this research is to enable early timing verification and feedback loops between verification and design of high-performance automotive systems. This research draws on and extends our framework for the development and architectural exploration of system-designs with temporal awareness described in [6].

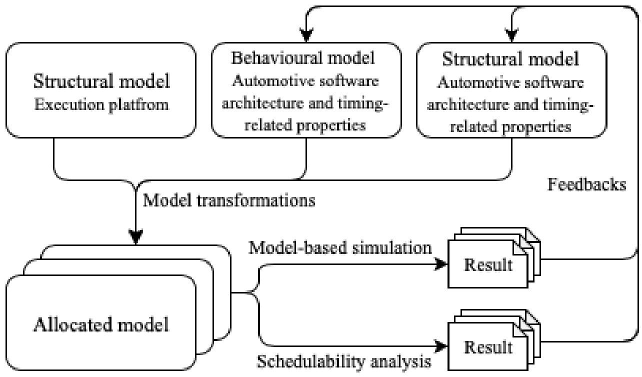

Fig. 1: Workflow of the proposed framework

Fig. 1 depicts the workflow of the proposed framework. In particular, the engineer will represent the automotive software architecture and its timing related properties in terms of platform-agnostic structural and behavioural models, first. The execution platform will be described in terms of structural models, too. Model transformations will automatically generate models with allocation information suitable for timing analysis and simulation. Schedulability analysis will provide early feedbacks on the timing compliance of the automotive software architecture while model-based simulation will provide feedbacks on its behaviour. As there might be several options for mapping software to hardware, models transformation might generate several models entailing unique and meaningful allocations. Eventually, the results of the schedulability analysis and the simulation will be handed to the engineer for enabling feedback loops between the verification and the design.

The remainder of this paper is as follows. Sec. 2 describes the the research line for this work. Sec. 3 presents existing related approaches documented in literature and compares them to our solution. Sec. 4 describes the steps we are plan to follow for carrying out this research effort.

2 Research Line

Towards the fulfilment of the research goal, we have identified the following research challenges (RCs) to be tackled.

RC1: Extension of modelling languages. Current industrial automotive modelling languages (e.g., EAST-ADL [10], RCM [5]) provide limited to no support for developing high-performance automotive systems. Timing verification suffers from this lack of support. Here the challenge is to extended these languages for allowing the modelling of automotive HPC platforms while ensuring backward-compatibility. Providing answers to this challenge is the key for enabling model-based timing verification.

RC2: Automation for timing verification. Current industrial timing verification techniques provide limited to no support for HPC computing platforms. What is more, they rely on allocated models. Here the challenge is to use model transformations for enabling automatic allocation of computations to computational units. Providing answers to this challenge will enable early timing-verification for high-performance automotive systems. This challenge can be broken down into two sub-challenges.

RC2.1: Automation for simulation-based assessment of timeliness. Here the challenge is to simulate HPC platforms and timing-related properties.

RC2.2: Automation for timing analysis. Here the challenge is to generate all the meaningful models entailing unique allocations of computations to computational units.

RC3: Automation for back-propagation. Early timing verification can be very effective if the results are used for enabling timing-aware design decision [11]. Here the challenge is to collect the timing verification results and to unveil them to the engineer. Providing answers to this challenge will enable timing verification in a continuous software development setting.

3 Related Work

In the last decades, a plethora of automotive-specific modelling languages and model-based methodologies for the software development of automotive systems have been proposed.

EAST-ADL is an architecture description language for the specification of automotive electronic systems [10]. It leverages a multi-layer approach, where each layer describes the vehicular system at a different abstraction level and from a different perspective. The last layer is usually delegated to domain-specific languages (DSL), notably AUTOSAR, RCM, and so forth. AUTOSAR was born as an industrial initiative to provide a standardised software architecture to EAST-ADL [9]. AUTOSAR has been continuously updated and refined: AUTOSAR 4.0 includes support for multi-core platforms. However, such a support mainly focuses on the adaptation of the AUTOSAR run-time support rather than allocation of computations to computational units. Currently, AUTOSAR does not support modelling of automotive systems on HPC platforms. Based on AUTOSAR, a number of frameworks for the specification of timing-analysis have been proposed [20] [12]. However, these frameworks do not support automatic allocation. What is more, these frameworks can not provide early precise timing analysis and enable feedback loops. Rubus Component Model (RCM) is a modelling language developed by Arcticus Systems AB and supporting the development of resource-constrained, real-time embedded systems [5]. Through the years, it has been mostly adopted within the vehicular domain by several national and international companies (e.g., Volvo Construction Equipment, BAE Systems, etc.). Currently, RCM does not support modelling for automotive systems on HPC platforms. Based on RCM, the work on [15] leverages high-precision timing analysis at system level. However, the approach is fully manual and the timing analysis only support homogeneous platforms.

Given the ubiquity of software, there exists a corpus of literature devoted to the design of embedded systems and posing a special focus to timing requirements. AADL is an architecture description language conceived for the avionic domain, but it has been increasingly used for modelling embedded systems in general [16]. Currently, AADL provides no support for automatic allocation and for the modelling of HPC platforms. Several research works are based on the use of general-purpose languages such as UML as alternatives to domain-specific languages. CHESS is a UML-based cross-domain approach for designing complex component-based embedded systems [8]. It allows to model a software system and its target platform(s). Moreover, it provides dedicated packages for specifying properties relevant to timing and dependability analysis. CHESS supports multi-core target platforms, but not HPC ones. GASPARD is a UML-based framework for the design of parallel embedded systems [4]. It prescribes a workflow made-up of subsequent analyses and refinement steps, from higher to lower abstraction levels. GASPARD focuses on complementary non functional properties than timing. In recent years, several approaches dealing with complex software systems development by adopting multi-paradigm modelling techniques and leveraging simulation mechanisms to perform early analysis of systems have been proposed [13]. However, they do not support HPC platforms.

There are several ongoing Swedish and European research projects addressing the software development of high-performance cyber-physical systems. DPAC is a Swedish research project which aims at providing dependable platforms for computer-controlled functionality in autonomous systems [1]. It targets software development and execution of embedded systems, but its main focus is dependability. PANORAMA is a European project focusing on model-based methods and tools for mastering the development of heterogeneous embedded hardware and software systems in the automotive and avionics domains [3]. HERO is a Swedish research project focusing at providing a framework for enabling development of optimised parallel software [2]. One of the main goals of HERO is to define a language for writing parallel software. However, this is complementary to the goal of this research effort.

4 Roadmap

We have divided this research effort into 6 project phases (PPs), as follows.

PP1 - Identification and specification of requirements and use cases. In this PP, we will work together with our industrial partners (e.g., Volvo Group Truck Technology and Arcticus Systems) for identifying the industrial requirements and use-cases for this research effort.

PP2 - Identification of modelling concepts for high-performance vehicular systems. In this PP, we will investigate the extensions of industrially relevant automotive-specific modelling languages (e.g., EAST-ADL and RCM) with concepts for representing the software architecture, timing-related properties, platform and allocation information of high-performance automotive systems. We will validate the proposed extensions using the automotive use cases identified in PP1. This PP will provide solutions for RC1.

PP3 - Automation for simulation-based assessment of timing. In this PP, we will work together with our industrial partner Volvo Group Truck Technology for extending their simulation framework. In particular, we will investigate the use of model transformations for the generation of simulation-ready models. Volvo Group Truck Technology will be responsible for the validation of the proposed extension. This PP will provide solutions for RC2.1.

PP4 - Automation for timing analysis. In this PP, we will work together with our industrial partner Arcticus Systems AB for extending their schedulability analysis framework with a plugin able of generating all the meaningful models entailing unique allocations. Arcticus Systems AB will be responsible for the validation of the proposed extension. This PP will provide solutions for RC2.2.

PP5 - Feedback loops. In this PP, we will investigate which information should be propagated back and how to unveil such an information to the engineer. This PP will provide solutions for RC3.

PP6 - Validation and dissemination of results. In this PP, we will carry out a set of cross-cut activities for the dissemination of the results as well as for the validation of the proposed methodology. In particular, the validation will use the use cases identified in PP1.

This research effort will span over two years from 1st April 2019 to 31st March 2021. PP1 will start in April 2019 and will have a duration of three months. PP2, PP3 and PP4 will have a duration of six months each and will start in June 2019, November 2019 and April 2020, respectively. PP5 will start in September 2020 and will have a duration of four months. The dissemination activities in PP6 will be carried out in correspondence of termination of each PP. While we are planning to have evaluation activities in each PP, the evaluation of the whole methodology will be done in PP6 and will have a duration of three months, from January to March 2021. We are not planning to run more than two technical PPs in parallel.

References

1. Dpac (2019), `http://www.es.mdh.se/projects/414-DPAC`
2. Hero (2019), `http://www.es.mdh.se/projects/511-HERO__Heterogeneous_systems___software_hardware_integration`
3. Panorama (2019), `https://itea3.org/project/panorama.html`

4. Abdoulaye, G., Sébastien, L.B., Éric, P., Rabie, B.A., Anne, E., Philippe, M., Dekeyser, J.L.: A model-driven design framework for massively parallel embedded systems. ACM Transactions on Embedded Computing Systems (TECS). 10(4), 39 ((2011))
5. Alessio, B., Antonio, C., Federico, C., Saad, M., Mikael, S.: Technology-preserving transition from single-core to multi-core in modelling vehicular systems. In: Springer (ed.) 13th European Conference on Modelling Foundations and Applications. ((2017)), http://www.es.mdh.se/publications/4750-
6. Alessio, B., Lorenzo, A., Antonio, C., Federico, C., Romina, E., Saad, M., Mikael, S.: Moves: A model-driven methodology for vehicular embedded systems. IEEE Access 6, 6424–6445 ((2018))
7. Bézivin, J.: On the unification power of models. Software & Systems Modeling 4(2), 171–188 (2005)
8. Cicchetti, A., Ciccozzi, F., Mazzini, S., Puri, S., Panunzio, M., Zovi, A., Vardanega, T.: Chess: a model-driven engineering tool environment for aiding the development of complex industrial systems. In: Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering. pp. 362–365. ACM ((2012))
9. Consortium, T.A.: Autosar techincal overview, version 4.3. ((2016)), http://autosar.org
10. EAST-ADL: East-adl domain model specification, deliverable d4.1.1. http://www.atesst.org/home/liblocal/docs/ATESST2_D4.1.1_EAST-ADL2-Specification_2010-06-02.pdf, accessed: 2019-03-20
11. Galin, D.: Software quality assurance: from theory to implementation. Pearson Education India (2004)
12. Goknil, A., DeAntoni, J., Peraldi-Frati, M.A., Mallet, F.: Tool support for the analysis of tadl2 timing constraints using timesquare. In: 2013 18th International Conference on Engineering of Complex Computer Systems. pp. 145–154. IEEE (2013)
13. Jensen, J.C., Chang, D.H., Lee, E.A.: A model-based design methodology for cyber-physical systems. In: Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International. pp. 1666–1671. IEEE (2011)
14. Lo Bello, L., Mariani, R., Mubeen, S., Saponara, S.: Recent advances and trends in on-board embedded and networked automotive systems. IEEE Transactions on Industrial Informatics 15(2), 1038–1051 (2019)
15. Mubeen, S., Nolte, T., Sjödin, M., Lundbäck, J., Lundbäck, K.L.: Supporting timing analysis of vehicular embedded systems through the refinement of timing constraints. Software & Systems Modeling pp. 1–31 (2017)
16. Peter H., F., David P., G., J.Hudak, J.: The architecture analysis & design language (AADL): An introduction. Tech. rep. ((2006))
17. Schmidt, D.C.: Guest Editor's Introduction: Model-Driven Engineering. Computer 39(2), 25–31 (Feb 2006)
18. Sendall, S., Kozaczynski, W.: Model transformation: The heart and soul of model-driven software development. Software, IEEE 20(5), 42–45 (2003), http://dx.doi.org/10.1109/MS.2003.1231150
19. Sha, L., Abdelzaher, T., Årzén, K.E., Cervin, A., Baker, T., Burns, A., Buttazzo, G., Caccamo, M., Lehoczky, J., Mok, A.K.: Real time scheduling theory: A historical perspective. Real-time systems 28(2-3), 101–155 (2004)
20. Stappert, F., Jonsson, J., Mottok, J., Johansson, R.: A design framework for end-to-end timing constrained automotive applications. Embedded Real-Time Software and Systems (ERTS) (2010)
21. Unibap: Intelligent Vision System. https://unibap.com/product/intelligent-vision-system/, accessed: 2019-03-20