Developing Predictable Embedded Systems in the Vehicle Industry: Results and Lessons Learned

Saad Mubeen
Mälardalen University, Sweden
saad.mubeen@mdh.se

***Abstract*—This paper discusses the results achieved in a technology-transfer project in the vehicle domain. The results extend the state-of-the art and practice in the area of model-driven development and execution of predictable vehicular embedded systems by developing new techniques, implementing them in the existing industrial tools and validating them using industrial use cases and prototype demonstrators. In this context, the paper also discusses some important lessons learned and open challenges.**

I. Introduction

Advanced features in modern vehicles require high levels of computation capabilities in Electronic Control Units (ECUs), mainly due to data-intensive sensors (e.g., camera and radar) and computation-demanding functionality [1], [2], [3]. The contemporary single-core ECUs are unable to keep up with these requirements, which are expected to increase driven by innovation in advanced vehicle architectures. To address these requirements, a promising solution has recently emerged with the introduction of multi-core processors that are specifically designed for vehicular applications, e.g., AURIX TriCore Family by Infenion[1], MPC5675K dual-core and MPC5777C tri-core by NXP[2]. Many vehicular embedded systems are required to be predictable due to their time-critical nature [4]. This means, it should be possible to prove or demonstrate at the design time that all specified timing requirements will be satisfied when the system is executed. In comparison to the single-core processors, multi-core processors are more prone to unpredictable behavior due to shared resources among the cores, e.g., system bus, I/Os and memories. Hence, development of time-predictable vehicular embedded systems on multi-core EUCs becomes a daunting challenge. PreView (Developing Predictable Vehicle Software on Multi-core) [5] is a technology-transfer project that takes on the above-mentioned challenge. This paper provides an overview of the results, experiences and lessons learned in the project.

II. PreView: Overview, Results & Lessons Learned

The goal of the Preview project (Aug 2016- Jul 2018) is to (i) develop a model-driven development methodology and techniques for predictable vehicular multi-core embedded systems, (ii) transfer the new techniques to the vehicle industry by implementing them in the existing industrial tools as well as developing new industrial tools, and (iii) demonstrate the usability of the techniques and tools on industrial prototypes.

[1]https://www.infineon.com/cms/en/applications/automotive
[2]https://www.nxp.com

PreView is a joint collaboration between Mälardalen University and three industrial partners Arcticus Systems[3], Volvo Construction Equipment[4] and BAE Systems[5] as shown in Fig. 1. The geographical separation among the project partners are in the range of [40-550] km. The project has a total of 17 associated members, out of which 12 are from the vehicle industry. This joint collaboration offers a clear value chain from academia, developer of the new techniques and methods; through the tool developer/vendor (Arcticus), implementer of the techniques in industrial tools; and finally, to the end users of the technology (Volvo and BAE), Original Equipment Manufacturers (OEMs), users of the existing tools (for over 20 years [6]) and the newly developed tools on prototype vehicles.

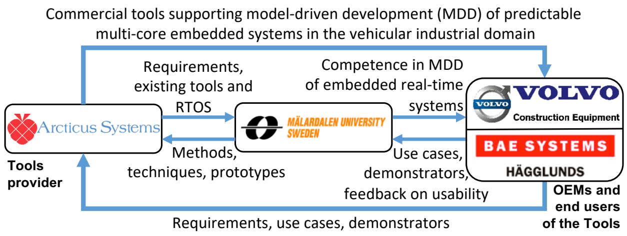

Fig. 1: Information flow and collaboration among the partners.

A. Inputs to the Project

Several inputs came from academia and industry at the start of the project, which are also shown on the left side in Fig. 2.

- The state-of-the-practice software development process for predictable vehicular embedded systems on single-core EUCs was one of the inputs to the project. This process allows to model the software architectures, verify time predictability, automatically generate code, deploy and run the systems on predictable single-core execution platforms.
- The Rubus tool chain was another input to the project. This tool chain complies with the above-mentioned development process. It has been used in the vehicle industry for over 20 years [6]. It consists of a complete Integrated Component model development Environment (ICE), designer, inspector, analyzer, code generator and simulator. It is also supported by a Real Time Operating System (RTOS) that has been certified according to ISO 26262 (ASIL-D) safety standard.

[3]https://www.arcticus-systems.com
[4]https://www.volvoce.com
[5]https://www.baesystems.com

- Over 20 years of experience of using the tool chain and the development process by Volvo and BAE was another input.
- Several existing scientific techniques for the timing analysis and execution of the systems on multi-core platforms.

B. Project Results

The project results extend the state-of-the art and practice in the area of model-driven development and execution of predictable vehicular embedded systems. The key scientific and industrial results, also depicted in Fig. 2, are as follows.

- One of the main results in the project is a model-driven development methodology to design temporally correct vehicular embedded systems on single- and multi-core ECUs [7].
- Another important result is the extension of the state-of-practice development process for single-core systems to support the development of predictable vehicular multi-core embedded systems as shown on the right side of Fig. 2.
- A new model- and component-based software development technique for multi-core vehicular embedded systems is developed and implemented in the existing industrial model, namely the Rubus Component Model (RCM).
- A new end-to-end timing analysis framework [8] is developed and implemented as the extended analysis engine for the Rubus Analyzer. The extended analyzer supports the verification of time predictability in vehicular multi-core embedded systems by analyzing their software architectures.
- Automatic code generation from the multi-core software architectures and corresponding extensions to the Rubus code generator.
- A new multi-core hypervisor that instantiates the certified Rubus RTOS for each core. It provides a predictable execution environment for multi-core ECUs.
- Validation and feedback on the usability of the above results by means of several use cases from the vehicle industry.

C. Experiences and Lessons Learned

The project provides valuable experience of transferring the state-of-art research results to the industry. Two of the most important lessons learned in the project are discussed below.

1) Predictability vs Performance: In this project, we witness a stand-off over predictability and performance in vehicular multi-core embedded systems. In order to support a predictable execution platform, the techniques implemented in the multi-core hypervisor allow the time and space partitioning among the cores. This means that the shared memories are partitioned, while the Time Division Multiple Access (TDMA) protocol is used to access the system bus. The TDMA protocol ensures predictability of the system, but hampers the system performance as the system bus can remain idle if a core does not need to access it during its designated time slot. The overall system performance reduces with the increase in the number of cores if TDMA is used to ensure predictability in multi-core systems. In PreView, we use the concept of partitions within cores to increase the performance (utilization of each core) without jeopardizing predictably, while using the TDMA protocol to access the system bus. For instance, an application that has performance requirements and no predictability requirement can be assigned to a non-critical partition on the same core, where another application with predictability requirements is allocated to a critical partition. This allows the critical application to preempt the non-critical application, while ensuring the best-effort performance for the non-critical application on the same core.

2) On-board Computation vs on-board Communication: In many vehicular applications that are distributed over more than one ECU, merely supporting powerful computation platforms in the form of multi-core ECUs (focus of PreView) is not enough. The on-board communication platforms (i.e., in-vehicle buses and networks) should equally support high data rates that are required by these applications; otherwise, the on-board communication becomes a performance and predictability bottleneck. Majority of the contemporary on-board networks such as CAN, LIN, Flexray and MOST do not meet the high data-rate requirements of modern data-intensive vehicular applications, especially in the autonomous vehicle domain. One notable exception is the set of IEEE AVB standards that support high-bandwidth on-board communication of up to 100 Mbit/s. However, AVB does not support low-latency communication, which is very important in ensuring predictability in time-sensitive vehicular applications that are distributed over several ECUs. The recently introduced IEEE TSN standards offer a promising solution to meet such requirements. However, these standards are in their infancy and require further research and academic-industrial collaborations to utilize TSN in the industrial applications. Hence, supporting model-driven development of computation- and data-intensive vehicular systems with requirements on predictability and high-bandwidth low-latency on-board communication remains an open challenge.

ACKNOWLEDGEMENT

The work in this paper is supported by the Swedish Knowledge Foundation via the projects PreView, HERO and DPAC and by the Swedish Governmental Agency for Innovation Systems (VINNOVA) via the project DESTINE. We thank all the industrial partners, especially Arcticus Systems, Volvo Construction Equipment and BAE Systems Hägglunds.

REFERENCES

[1] G. Gut, C. Allmann, M. Schurius, and K. Schmidt, "Reduction of electronic control units in electric vehicles using multicore technology," in *International Conference on Multicore Software Engineering, Performance, and Tools*, 2012, pp. 90–93.

[2] D. Reinhardt and M. Kucera, "Domain controlled architecture - a new approach for large scale software integrated automotive systems," in *International Conference on Pervasive and Embedded Computing and Communication Systems (PECCS 2013)*, February 2013, pp. 221–226.

[3] L. Lo Bello, R. Mariani, S. Mubeen, S. Saponara, "Recent advances and trends in on-board embedded and networked automotive systems," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, 2019.

[4] D. Grund, J. Reineke, and R. Wilhelm, "A Template for Predictability Definitions with Supporting Evidence," in *Bringing Theory to Practice: Predictability and Performance in Embedded Systems*, ser. OpenAccess Series in Informatics, vol. 18, Dagstuhl, Germany, 2011, pp. 22–31.

[5] PreView Project: Developing Predictable Vehicle Software on Multi-core, http://www.es.mdh.se/projects/442-PreView, accessed Oct., 2018.

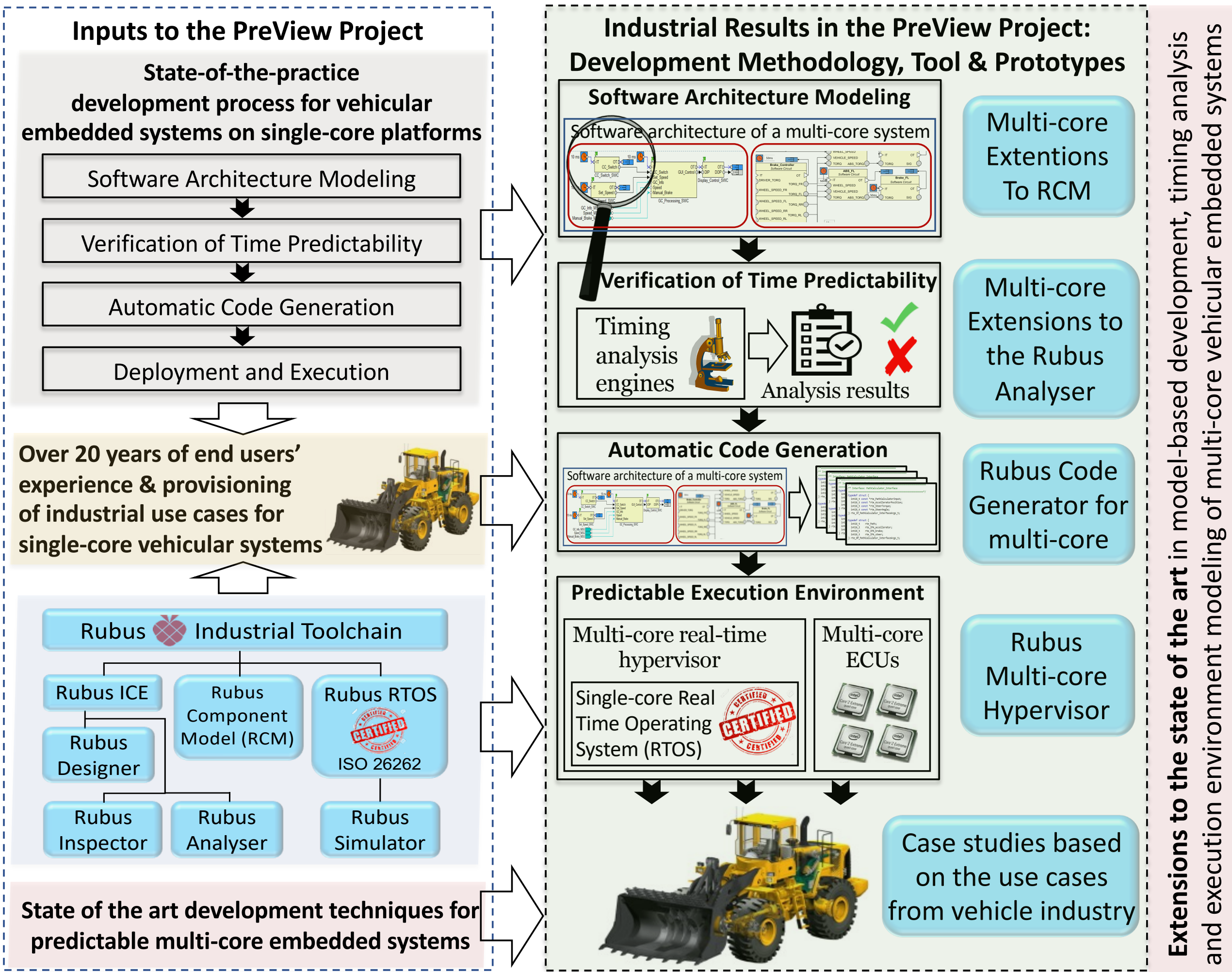

Fig. 2: PreView project: overview of the inputs, development methodology, scientific and industrial results.

[6] S. Mubeen, H. Lawson, J. Lundbäck, M. Gålnander, and K. L. Lundbäck, "Provisioning of Predictable Embedded Software in the Vehicle Industry: The Rubus Approach," in *IEEE/ACM 4th International Workshop on Software Engineering Research and Industrial Practice*, 2017, pp. 3–9.

[7] A. Bucaioni, L. Addazi, A. Cicchetti, F. Ciccozzi, R. Eramo, S. Mubeen, and M. Sjödin, "Moves: A model-driven methodology for vehicular embedded systems," *IEEE Access*, vol. 6, pp. 6424–6445, 2018.

[8] M. Becker, S. Mubeen, D. Dasari, M. Behnam, and T. Nolte, "A generic framework facilitating early analysis of data propagation delays in multi-rate systems (invited paper)," in *23rd IEEE International Conference on Embedded & Real-Time Computing Sys. & Applications*, 2017, pp. 1–11.