Chapter 2

FORMAL METHODS FOR INTEGRATION OF AUTOMOTIVE SOFTWARE

Marek Jersak¹, Kai Richter¹, Razvan Racu¹, Jan Staschulat¹, Rolf Ernst¹, Jörn-Christian Braam² and Fabian Wolf²
¹ Institute for Computer and Communication Network Engineering, Technical University of Braunschweig, D-38106 Braunschweig, Germany; ² Aggregatelektronik-Versuch (Power Train Electronics), Volkswagen AG, D-38436 Wolfsburg, Germany

Abstract. Novel functionality, configurability and higher efficiency in automotive systems require sophisticated embedded software, as well as distributed software development between manufacturers and control unit suppliers. One crucial requirement is that the integrated software must meet performance requirements in a certifiable way. However, at least for engine control units, there is today no well-defined software integration process that satisfies all key requirements of automotive manufacturers. We propose a methodology for safe integration of automotive software functions where required performance information is exchanged while each partner’s IP is protected. We claim that in principle performance requirements and constraints (timing, memory consumption) for each software component and for the complete ECU can be formally validated, and believe that ultimately such formal analysis will be required for legal certification of an ECU.

Key words: automotive software, software integration, software performance validation, electronic control unit certification

1. INTRODUCTION

Embedded software plays a key role in increased efficiency of today’s automotive system functions, in the ability to compose and configure those functions, and in the development of novel services integrating different automotive subsystems. Automotive software runs on electronic control units (ECUs) which are specialized programmable platforms with a real-time operating system (RTOS) and domain-specific basic software, e.g. for engine control. Different software components are supplied by different vendors and have to be integrated. This raises the need for an efficient, secure and certifiable software integration process, in particular for safety-critical functions.

The functional software design including validation can be largely mastered through a well-defined process including sophisticated test strategies [6]. However, safe integration of software functions on the automotive platform requires validation of the integrated system’s performance. Here, non-functional system properties, in particular timing and memory consumption are
the dominant issues. At least for engine control units, there is today no established integration process for software from multiple vendors that satisfies all key requirements of automotive OEMs (original equipment manufacturers).

In this chapter, we propose a flow of information between automotive OEM, different ECU vendors and RTOS vendors for certifiable software integration. The proposed flow allows to exchange key performance information between the individual automotive partners while at the same time protecting each partner’s intellectual property (IP). Particular emphasis is placed on formal performance analysis. We believe that ultimately formal performance analysis will be required for legal certification of ECUs. In principle, analysis techniques and all required information are available today at all levels of software, including individual tasks, the RTOS, single ECUs and networked ECUs. We will demonstrate how these individual techniques can be combined to obtain tight analysis results.

2. CURRENT PRACTICE IN SOFTWARE INTEGRATION

The software of a sophisticated programmable automotive ECU, e.g. for power train control, is usually composed of three layers. The lowest one, the system layer consists of the RTOS, typically based on the OSEK [8] automotive RTOS standard, and basic I/O. The system layer is usually provided by an RTOS vendor. The next upward level is the so-called ‘basic software’ which is added by the ECU vendor. It consists of standard functions that are specific to the role of the ECU. Generally speaking, with properly calibrated parameters, an ECU with RTOS and basic software is a working control unit for its specific automotive role.

On the highest layer there are sophisticated control functions where the automotive OEM uses its vehicle-specific know-how to extend and thus improve the basic software, and to add new features. The automotive OEM also designs distributed vehicle functions, e.g. adaptive cruise-control, which span several ECUs. Sophisticated control and vehicle functions present an opportunity for automotive product differentiation, while ECUs, RTOS and basic functions differentiate the suppliers. Consequently, from the automotive OEM’s perspective, a software integration flow is preferable where the vehicle function does not have to be exposed to the supplier, and where the OEM itself can perform integration for rapid design-space exploration or even for a production ECU.

Independent of who performs software integration, one crucial requirement is that the integrated software must meet performance requirements in a certifiable way. Here, a key problem that remains largely unsolved is the reliable validation of performance bounds for each software component, the whole ECU, or even a network of ECUs. Simulation-based techniques for performance validation are increasingly unreliable with growing application and architecture complexity. Therefore, formal analysis techniques which consider