



Technology Transfer in Computing Systems

D3.36: Individual TTP36 abstract

Project no.: 609491
Funding scheme: Collaborative project
Start date of the project: 1st September 2013
Duration: 36 months
Work programme topic: FP7-ICT-2013-10

Deliverable type: Report
Deliverable reference number: ICT-609491 / D3.36
WP and tasks contributing: WP 3 / all
Due date: 30/06/2016
Actual submission date: 27/07/2016

Responsible Organization: TUDENMARK
Dissemination Level: Public
Revision: 1.0



AUTOMAP 36: AUTOMAP: Tool for automatic mapping of AUTOSAR runnables to multi-core automotive architectures

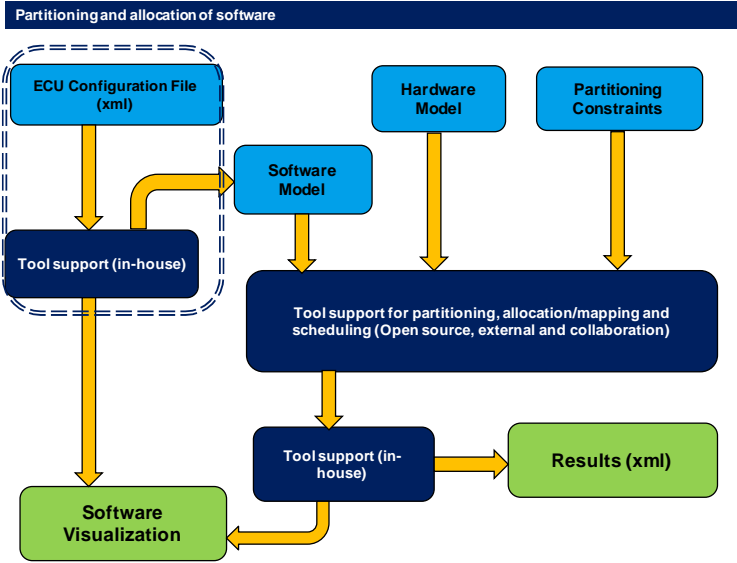
Paul Pop, Florin Maticu, Cosmin Avasalica, Dhanesh Budhrani Budhrani (Technical University of Denmark, Kongens Lyngby, Denmark), Axbrink Christian, Islam Mafijul, Ekelin Cecilia, Söderqvist Thomas (AB Vovlo, Gothenburg, Sweden)

The automotive electronic architectures have moved from federated architectures, where one function is implemented in one ECU (Electronic Control Unit), to distributed architectures, where several functions may share resources on an ECU. In addition, multicore ECUs are being adopted because of better performance, cost, size, fault-tolerance and power consumption.

The assignment of functions to the distributed vehicle architecture (also called “mapping”) is currently done manually. However, such a manual mapping is no longer feasible due to the introduction of multicores, which increases the complexity of the decisions and their impact, the use of complex middleware, such as AUTOSAR (AUTomotive Open System ARchitecture), and the required compliance to the functional vehicle safety standard ISO 26262. Without an automatic mapping solution, it is very challenging to utilize multicore-based ECUs in automotive systems, as experienced by Volvo Group, one of the leading manufacturers of commercial vehicles.

In this TTP, we are focusing on designing and developing a method and tool for deciding the assignment of functionality in automotive applications to multicore distributed architectures, consisting of several ECUs interconnected by a bus such as Controller Area Network. The platform details are given by Volvo, and comply with AUTOSAR and ISO 26262. An AUTOSAR application is composed of a several *software components* that interact with each other through well-defined *ports*. The implementation of a software component does not depend on a particular hardware or type of communication needed for sending signals, therefore it can be relocated and run on different ECUs. The behavior of a software component is constructed using the entities called *runnables*. They are software functions that implement the algorithms (behavior) of a software component. Once a software component is mapped to an ECU, the configuration and generation of the Runtime Environment (RTE) is applied. The AUTOSAR Operating System is based on the industry standard OSEK OS, and we assume that a fixed priority fully preemptive scheduling policy is used. The unit of execution inside AUTOSAR OS is called an *Os-Task*. Each *Os-Task* has assigned a priority and it can always be preempted by another *Os-Task* with a higher priority value. An *Os-Application* is an AUTOSAR entity that groups together a collection of *Os-objects* defined as *Os-Tasks*, *Interrupt Service Routines*, *alarms*, *events*, *counters*, etc.

The tool flow is presented in the figure below. Based on a hardware model, a software model and configuration information an allocation meeting a set of constraints is determined. The tool can also generate graphs for visualization of various resulting information.



Given an application model and an architecture model the mapping tool determines: a mapping of software components to ECUs, a mapping of runnables to cores, a mapping of runnables to OS-Tasks, a mapping of OS-Task to OS-Applications. The mapping attempts to minimize the overall communication bandwidth, the variance of core utilization of the system such that mapping constraints, if specified, are satisfied and the runnables are schedulable.

The tool is developed in C# using the .NET framework. The tool can be configured to take into account several inter-core and inter-ECU communication mechanisms, such as spin-locks, shared memory, and also the AUTOSAR RTE. The tool uses a "Simulated Annealing" meta-heuristic, which produced an output within 8 minutes, which satisfies the end-to-end signal constraints provided by the designer.

The tool is mature and it is currently extensively tested by Volvo. We also plan to use the tool to evaluate several inter-core communication mechanisms. The tool has been used on a Volvo Use Case. The application model consists of 50 software components with 75 runnables in total. The platform model assumes a communication model using spin-locks and consists of one ECU with 3 cores.

Efficient utilization of multicores and compliance with functional safety standard ISO 26262 are among the key business needs and challenges while designing the next generation of architectures for commercial vehicles. This TTP will contribute significantly to meet these needs by providing an efficient method and tool to harvest the full potential of multicores. Based on this, Volvo will be able to offer new functionalities and features to the customers through higher integration and better performance by improving resource utilization. Furthermore, this will contribute to the development of safer and more environment-friendly vehicles aligning thus with Volvo's vision of becoming world leader in sustainable transport solutions. The developed method and tool assist in designing and developing multicore-ready software systematically within Volvo. The expectation is that the tool will allow Volvo to reduce the costs (by using multicores and reducing the number of ECUs), maximize performance and resource utilization and handle the increased software complexity.