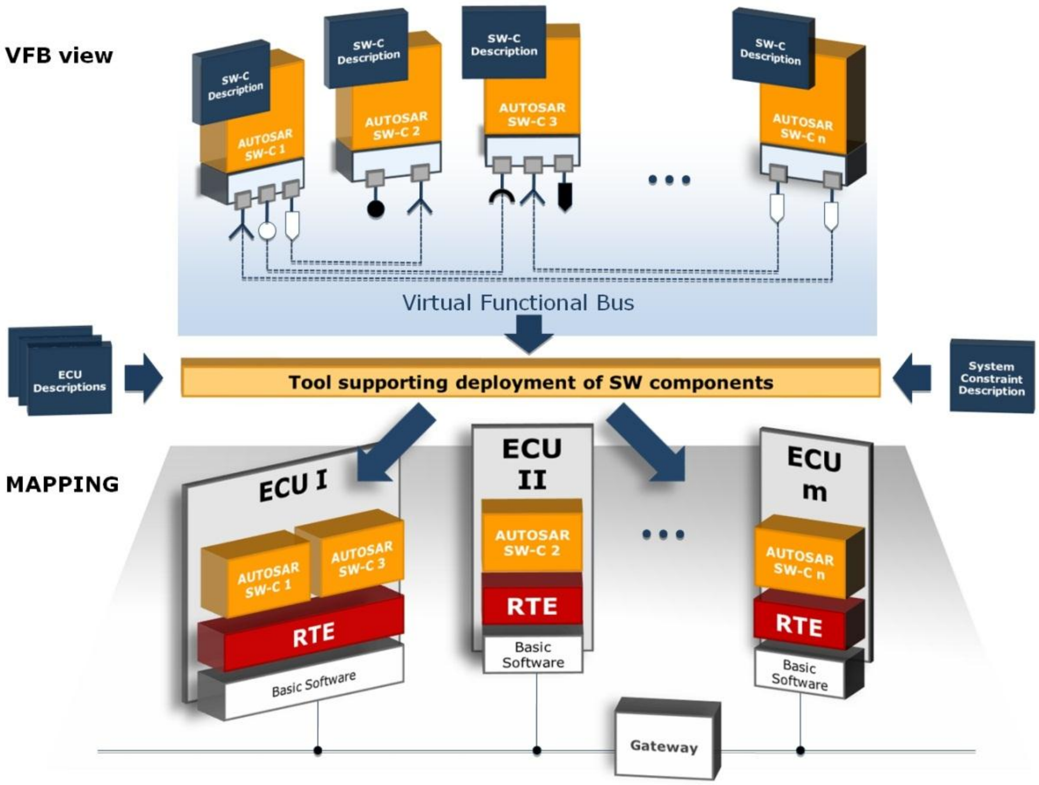
# AUTOSAR for Multi-Core in Automotive and Automation Industries



Deployment of AUTOSAR software on ECUs [2].

Project within the "Möjliggörande Elektronik" program

Kristian Sandström, ABB AB
Michael Svenstam, ArcCore AB
Björn Lisper and Thomas Nolte, Mälardalen University
Anders Svensson, Volvo Technology AB

December 2014

## Content

### FFI in short

FFI is a partnership between the Swedish government and automotive industry for joint funding of research, innovation and development concentrating on Climate & Environment and Safety. FFI has R&D activities worth approx. €100 million per year, of which half is governmental funding. The background to the investment is that development within road transportation and Swedish automotive industry has big impact for growth. FFI will contribute to the following main goals: Reducing the environmental impact of transport, reducing the number killed and injured in traffic and Strengthening international competitiveness. Currently there are five collaboration programs: **Vehicle Development, Transport Efficiency, Vehicle and Traffic Safety, Energy & Environment and Sustainable Production Technology.**

**For more information:** www.vinnova.se/ffi

# 1. Executive summary

The processing power of computer CPUs is continuously increasing, bringing the potential for additional features and more complex computer operations in industrial systems. Though, the processing power have shifted from increasing speed of a single core CPU to an increasing number of cores, each core typically not substantially more powerful than previous generation in terms of clock speed. This shift from single-core to multi-core CPU architectures has already taken place in many areas, e.g., desktop PCs, telecom systems, and even mobile devices. In some industrial domains, like automotive and industrial automation, this shift is now happing. One reason for this is that, for many processor families, the new generations of CPUs only come as multi-core architectures. Hence the only economically viable option for new generations of industrial controllers is to adopt these new multi-core architectures.

To be competitive the processing power of the new CPUs need to be fully utilized. Although, for the domains of automation and automotive, predictability is a top priority and have to be kept as well. New challenges with respect to predictability arise when the software can be executed truly in parallel on multiple concurrent processing units. One example of such a new challenge would be to find an optimal allocation of software components to CPU cores, considering both the parallel execution and cost for intra-core versus inter-core communication.

The new multicore architectures are more complex than the architectures they are replacing, but they also have the potential for innovation and optimization in industrial systems. For example, multicore CPUs can be made more energy efficient. Also, the cores in the CPU act as multiple separate computing units, opening up the possibility for hardware consolidation. The latter can be realized when several electronic computing units are replaced by a single multicore unit, moving the software of each unit to a separate core. Managing such consolidation, without risk for future increase in software development efforts, require adequate support by the software architecture with respect to software independence and integration. The AUTOSAR standard aims at providing a software architecture for integration of software components that is independent of component location in a distributed computer control system. This make AUTOSAR a good candidate architecture for developing concepts related to integration also in multicore based system.

The project AUTOSAR for Multi-Core in Automotive and Automation Industries aims at developing architectural concepts, analysis methods, and optimization techniques that achieve resource efficient and predictable usage of multi-core technology in the context of AUTOSAR. Specific goals have been to target challenges related *predictability* and *performance*, which must be solved in order to make efficient use of multi-core technology in embedded software applications. A third goal has been to use the new technology for innovation by finding software architectures and platform concepts for exploiting capabilities for *hardware consolidation* and *software integration*.

The results from the project in terms of conference and workshop publications, thesis reports, and technical reports are manifold and address the different goals. For supporting the goal of *predictability,* analysis methods have been developed for meeting end-to-end timing requirements in automotive and automation systems. For supporting *performance,* optimization techniques have been developed for task allocation in multicore systems, and also analysis and optimization of AUTOSAR run-time system mechanisms for use with multicore CPUs. For supporting *integration and hardware consolidation*, platform concepts have been developed for partitioning software on multicore CPUs and for integrating software components in a distributed multicore system.

It can be concluded that many of the challenges of introducing multicore CPUs are shared between automotive and industrial automation and, although the desired implementation frameworks sometimes may differ, many of the founding principles for the solutions can be shared between the two domains, and likely also by other domains.

# 2. Background

Over the past decade multi-core technology has developed to be the de-facto processor technology. This put us in a situation where in the near future only multi-core processors are available on the market, and therefore we have no choice but to use multi-core processors even when not required by the application. For example, the latest generation processor family from Tilera features devices with 16 to 100 cores [40]. Few software applications are developed for parallel architectures, so there will be major problems when making the shift in technology and migrating to multi-core processors. Issues related to **predictability** and **performance** must be solved in order to make efficient use of multi-core technology in embedded software applications [1].

However, being forced to use multi-core processors is not necessarily only a bad thing. For example, multi-core technology provides more powerful processors which opens up for **hardware consolidation** where it is possible to replace several previously dedicated single processors with one shared multi-core processor, thereby reducing the overall hardware and electronics cost of the system. Moreover, new multi-core technology yields more computing power, also providing new possibilities for innovation.

Concurrently, the automotive industry has developed a common software architecture for automotive embedded software systems. AUTOSAR (AUTomotive Open System ARchitecture) is a worldwide development partnership of car manufacturers, suppliers and other companies from the electronics, semiconductor and software industry [2]. Volvo for example has based the development of its next generation of trucks on AUTOSAR which means that the standard will be very important in the future.

The standard has recently been extended to support multi-cores, and such support at the system level is optional in AUTOSAR 4.0 [17]. The AUTOSAR consortium estimates that the production of electronic control units (ECUs) running AUTOSAR to about 25 million units in 2011 and that 220 million units will be sold across all AUTOSAR versions in 2016 [2].

The AUTOSAR software architecture shares many elements with those found in other application domains, including industrial automation. Therefore it is believed that AUTOSAR can be applied in many segments also outside the automotive domain.

The AUTOSAR partnership has now opened up the possibility to use the standard in non-automotive application domains [2]. This provides an opportunity for companies in other domains to take advantage of this worldwide standardization effort.

To tackle the challenges when moving to multi-cores and AUTOSAR we believe that cooperation between companies in different industrial segments will be essential, as well as cooperation with academic research groups targeting the relevant issues. Thus, the project consortium has partners both from automotive, industrial automation, and academia.

## Related Work

### Consolidation

Many industrial segments are very cost-aware, which implies a continuous focus on the product cost. Moreover, predictability, while maintaining high performance, is a requirement for industrial customers. By moving different functionality into a single multi-core hardware unit, thus eliminating hardware units, considerable cost cuts can be made.

Software consolidation is facilitated using the AUTOSAR approach where software components are executing on a common software architecture specified by the AUTOSAR standard. Hence, AUTOSAR is an important step towards consolidation of previously federated systems consisting of interconnected nodes.

Although AUTOSAR has been developed with automotive applications in focus, other industrial domains face many of the same challenges. It is therefore important to understand commonalities and differences, with respect to the AUTOSAR standard, between automotive systems and other application domains such as industrial automation.

A promising technique to achieve hardware consolidation is virtualization, which creates a virtual computer for the software application. This decoupling of physical resource from software significantly simplifies hardware consolidation, independence and integration on software on a common hardware platform. Implementations can be found as, e.g., hypervisor solutions. A central feature of these implementations is that they follow a hierarchical scheduling scheme, where the system resource is scheduled to partitions and, in turn, these partitions are scheduled locally for the users of the respective partitions.

Hierarchical real-time scheduling, originating in open systems in the late 1990's, has received an increasing research attention over the past years. Since Deng and Liu [3] introduced a 2-level hierarchical scheduling framework, its schedulability has been analyzed [4][5]. Shin and Lee [6] introduced the periodic resource model (to characterize the periodic resource allocation behaviour), and many studies have been proposed on schedulability analysis with this resource model [7][8][9][10]. The current research efforts are focused on multiprocessors, using techniques similar to those that have been used in the single processor case [11][12][13].

**Predictability**

Research in timing predictability on multi-core processors consider (1) how to schedule the software execution on multi-core platforms such that timing guarantees can be provided, and (2) techniques to safely estimate execution times for software executing on such platforms.

Two central approaches for scheduling of real-time systems on multiprocessors exist [14][15]; global and partitioned scheduling. Under global scheduling protocols, tasks are scheduled by a single scheduler and each task can be executed on any processor. A single global queue is used for storing tasks. A task can be pre-empted on a processor and resumed on another processor, i.e., migration of tasks among cores is permitted. Under a partitioned scheduling protocol, tasks are statically assigned to processors and the tasks within each processor are scheduled by a uniprocessor scheduling protocol. Each processor is associated with a separate ready queue for scheduling task jobs. There are systems in which some tasks cannot migrate among cores while other tasks can migrate. For such systems neither global or partitioned scheduling methods can be used. A two-level hybrid scheduling approach [16], which is a mix of global and partitioned scheduling methods, is used for those systems.

The automotive industry generally prefers static algorithms because of their more predictable behavior. The specification of the AUTOSAR operating system (OS) is based on the OSEK/VDX OS. Static priorities are assigned to tasks off-line and the highest priority first policy is used, with FIFO as a second criterion when more than one task has the same priority. The scheduling can be full non-preemptive, full preemptive or mixed preemptive (which means that every task can be preemptive or non-preemptive). Only uniprocessor systems are addressed in AUTOSAR release 3.2. The standard has recently been extended to support multi-cores and such support at the system level is optional in AUTOSAR 4.0 [17]. The objective of the extension has been to have minimal changes to the existing single-core system. Over the years a number of performance studies have been performed for multi-core and multiprocessor systems, hinting on which design decisions that must be made in order to achieve a resource efficient solution. Studies include performance of partitioned multi-core real-time systems [18], performance on soft real-time systems on multiprocessors [19], and performance when it comes to real-time systems containing dependencies [20].

Scheduling in an AUTOSAR context has received some attention in the literature. One of the findings in [21] is that real-time scheduling theory could be used for the analysis of real-time applications built upon the scheduling service provided by an AUTOSAR OS. The work presents a review of a subset of the AUTOSAR OS specification from a scheduleability point-of-view. The work presented in [22] and [23] proposes a static partitioning of the tasks on multi-cores and then schedule the tasks allocated on each ECU core based on table-driven periodic cyclic scheduling. In [24] the focus is on scheduling analysis techniques of real-time systems deployed on AUTOSAR compliant architectures. The paper studies the extent to what the AUTOSAR system model allows specifying necessary information that enables the model to be analyzed from a scheduling point of view. A concrete AUTOSAR system is evaluated using an open source tool, MAST [25].

Real-time scheduling requires safe estimates of the worst-case execution times (WCETs), obtained by a WCET analysis, for the tasks in the system. WCET analysis for multi-cores is very difficult, and a topic of active research. The major problem is that shared resources, like shared memories and buses, are now being accessed concurrently by concurrent activities on different cores which make them affect each others' timing. This interaction can be very complex which means that on code level, the timing predictability of multi-core processors can easily become very low. Subsequently, safe WCET estimates may become very imprecise. For that reason, almost all research on WCET analysis for multi-cores assume a TDMA arbitration policy for the system on-chip bus which gives each core a guaranteed bandwidth to memory: see, for instance, [26][27].

It is evident that the scheduling policy will affect the WCET's of tasks, since the scheduling decides which tasks will possibly run concurrently and thus affect each others' timings. Since, conversely, the WCET's will affect the scheduling policies, WCET analysis and scheduling policies cannot be considered in isolation for multi-cores. There is very little work done in this area, one of the few examples is [28] where a precedence relation between tasks is used to restrict possible task interactions affecting the L2 instruction cache.

**Performance**

Performance issues are central in industrial applications. The project will address performance issues at the node and system levels.

**a) Node performance**

In the AUTOSAR standard, the application is divided into Software Components. The Run-Time Environment relies on the AUTOSAR Operating System and Basic Software (BSW). The BSW forms the biggest part of the standardized AUTOSAR environment [2].

The AUTOSAR multi-core OS architecture specification states that “All BSW modules that would be present on a Single-core system usually reside on the master core in a multi-core system” [17]. As the BSW constitutes a significant fraction of the software, restricting it to one core can severely limit the performance potential of multi-cores. Recent studies have shown that the ECU performance actually becomes worse with multi-core in the AUTOSAR environment [29]. The cited study proposes a different approach, using a simple spin-lock that is taken whenever the BSW is entered. Their measurements show that this simple approach performs significantly better than the AUTOSAR approach.

**b) System performance**

Traditionally, automotive embedded systems are built up using a large number of ECUs with hardware specifically tailored to their particular functionality. The heterogeneity of this system design has a number of drawbacks:

- Inefficient use of computational power. Some ECUs will be pushed to their limits, while other will idle. Functional growth will be sacrificed as the hardware cannot support it.
- Slow evolution of functionality. With each ECU being tailor-made, all software for the ECU needs to be too. Subsequent changes of the ECU setup will require extensive rework of the software.
- Complex maintenance. Every ECU will have its own failure characteristics.
- Many different HW components. This brings issues like managing a large amount of part numbers, inventory, after market considerations, etc.

With the introduction of AUTOSAR, there are now opportunities to make radical changes in how the automotive embedded systems are structured and distributed. One can envision a system consisting of a number of generic computational ECUs with high-performance multi-core processing and memory resources, connected in a high-speed network, with specialised I/O nodes. This is the route that the avionics industry has been on since for many years. The introduction of IMA (Integrated Modular Avionics) has enabled a move towards distributed electronic systems where powerful and generic computational nodes are connected to smart I/O-nodes in high-speed networks [32]. Such an architecture yields cost savings through fewer types of hardware, increased robustness with the ability to dynamically adapt to faults, high scalability by adding standard ECUs, and high evolvability since it becomes easier to introduce advances in embedded electronics into the ECUs.

The platform based design methodology [33][34][35][36][37] is a very promising approach to investigate how future automotive electrical architectures could evolve into something similar to that used in avionic systems today. The computational ECUs will be based on multi-core processors. The smart I/O nodes contain extremely limited functionality and instead all logic is placed in the main ECUs. The methodology uses clearly identified abstraction layers and a design interface that allows for separation of concerns between refinement of the functional architecture specification and the abstractions of possible implementations. The application space is represented by a platform-independent vehicle function model. The architecture space is represented by an execution architecture model which is independent of functionality. The platform abstraction is represented by a system platform model which is independent of both. In automotive AUTOSAR systems the application space is represented by interconnected SWCs, the architecture space is represented by ECUs and communication networks and the platform abstraction is represented by tasks and messages. Design space exploration consists of finding the system platform model’s optimal mapping into the candidate execution platform instances.

# 3. Objective

The overall objective of this project is to achieve resource efficient and predictable usage of multi-core technology in the context of AUTOSAR. Doing so, we will explore and show the concrete benefits and potential of multi-core and also to understand its limitations.
Relying on the context of AUTOSAR, the intended application domain of the solutions developed in this project is broadened from a traditional exclusive automotive application to also include AUTOSAR for industrial automation. With the two application domains, complementary requirements and experiences of the project members will guide the research and development in this project.

**Project Challenges**
In order to achieve the overall goal of this project we have identified a set of challenges that must be addressed:

**Hardware consolidation**
A common drive for the introduction of multi-core technology is the need for hardware consolidation. By integrating software that previously was executing on its own dedicated hardware, on shared multi-core hardware, substantial savings can be made with respect to hardware cost, and software complexity. Hence, there is a huge potential in consolidating hardware and integrating software on a common multi-core platform (single multi-core, or a set of interconnected multi-cores).

- A key challenge in this project is to identify and develop techniques related to software architecture to allow for hardware consolidation.

**Predictability**
A key requirement for automotive and automation applications is predictability. However it is not enough to achieve a predictable functional behavior of the software, it must also be predictable with respect to non-functional requirements such as timing. Hence, the targeted software systems should be considered as real-time systems.

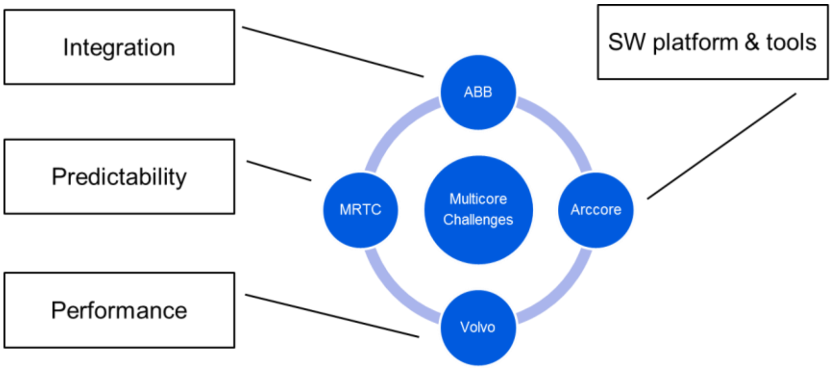
- A key challenge in this project is to develop techniques to achieve timing predictability of software executing on multi-core platforms.

**Performance**
Resource efficiency is central to the usefulness of multi-core technology in automotive and automation applications. Hence, a solution that does not solve the performance challenge is not sufficient for a broad adoption in real industrial applications.

- A key challenge in this project will be to develop resource efficient techniques for the execution of AUTOSAR software on multi-cores.

# 4. Project realization

The project is centered on challenges related to introduction of multicore platforms in industrial system. A set of research areas, as depicted in the table below, was defined at project inception and this has been carried through to the end of the project. These areas are separate but interplay to create successful industrial solutions. Each actor in the project consortium has had its own area of responsibility and the results have been transferred continuously through joint project meetings, workshops, and disseminations. In order to validate the results each area has developed a lightweight dedicated evaluation framework for that specific topic. The evaluation platforms themselves and the results produced using the platforms have been disseminated through project meetings, thesis reports, publications, and reports. Special effort have been put into discussing commonalities and differences between the different domains of automotive and automation.



The project is divided into a number of work packages as shown in the table below.

| |
|---|
| **WP1 Hardware Consolidation – Leader: ABB** |
| This work package has the responsibility for providing techniques that allow for hardware consolidation enabling existing previously federated system architectures to be integrated in a shared architecture hosting several software applications in parallel.<br>We will cover research in software independence in the context of multi-core architectures. If several hardware nodes are consolidated on a single multi-core CPU, it is important to keep the independence between different software components with respect to e.g., real-time behaviour and verification. Moreover, when several multi-core nodes are interconnected with a network, components should be allocated to cores and nodes based on well-founded methods. |
| **WP2 Predictability – Leader: MDH** |
| This work package has the responsibility for providing techniques that guarantee timing predictability of software execution. Techniques include scheduling techniques for multi-core, schedulability analysis techniques for multi-core, and execution time analysis techniques for multi-core. Moreover, we will look at solutions that combine scheduling and execution time analysis. |
| **WP3 Performance – Leader: Volvo** |
| This work package has the responsibility for ensuring performance in the techniques that are developed in the project, including system wide performance. This includes system configuration (e.g., parameter settings), system partitioning e.g., how to distribute software parts in the system, on the nodes and on the cores, and which run-time techniques to use to maximize performance. |
| **WP4 Demonstrator – Leader: ArcCore** |
| This work package is responsible for the project demonstrator, where we integrate techniques developed in the project into a tool suite such that new systems can be developed showing the (possible) potential in the technical solutions. The software and hardware platforms and necessary tools for the demonstrator are defined here. |

# 5. Results and deliverables

This chapter presents project results and deliveries which have been divided into a number of parts depending on the main challenges addressed: hardware consolidation and integration (Section 5.1), predictability (Section 5.2) and performance (Section 5.3).

## 5.1 Results related to challenge hardware consolidation and integration

One of the goals of the project has been to determine whether AUTOSAR is capable of meeting the technical needs of the industrial automation domain. In order to answer to this question, we have made a study the of AUTOSAR standard and the commonalities and differences between the automotive and industrial automation domains. Furthermore we have looked into the key features and capabilities of the AUTOSAR standard which are valid for the industrial automation domain. This has been reported in [pub6].

A set of the technologies identified in the initial study has been conceptualized in an industrial automation context. An evaluation framework relevant for the industrial domain was defined and developed, as reported in [pub1] [pub13], and two different concepts have been developed and evaluated as reported in [pub12][pub14]. The concepts also include theories for formal analysis of the system behavior.

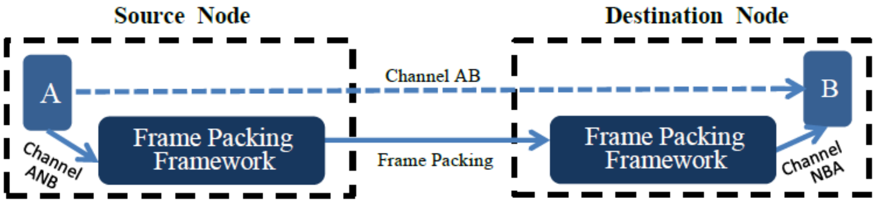


Figure1: Frame Packing Framework.

**Summary of Deliverables**

- An evaluation framework based on industrial automation requirements [pub1][pub13].
- Concept for Industrial automation VFB including model and solution for frame packing as seen in Figure 1, reported in [pub12].
- A definition of signal specification, where a signal is a data that is sent from one software component and received by another, possibly remote, software component with a requirement on timely delivery.
- A mapping of a set of signals with individual timing requirements, sizes, and destinations onto a set of network frames where signal origins and destinations can be identified.
- A framework for time triggered distribution of network frames using Ethernet, between a set of computer nodes, according to predefined schedules.
- An analysis model for calculating latency of transmission of frames on a switched Ethernet including the software stack at receiving and sending end.
- Partitioning model, deployment and QoS, overview and comparison [pub7].
- A concept framework for integration of software partitions on multicore with support for QoS [pub14].

## 5.2 Results related to challenge *predictability*

One of the challenging aspects for shifting AUTOSAR-based applications from single-core to multi-core platforms is how to achieve a predictable design while at the same time utilizing the system efficiently. The main reason complicating this problem is communication between the runnables. It is desirable to minimize the inter-runnable communication cost in order to increase system efficiency. Both mapping of runnables to tasks and allocation of the tasks among processing cores significantly affect the communication cost. Accordingly, introducing a framework which covers both the process of mapping runnables to tasks and the allocation of tasks to cores with communication reduction considerations is highly valuable, which in turn has the potential to increase system efficiency. Experiments have shown that this proposed framework is more effective for a highly connected set of runnables where communication between them is not too small. Nevertheless, for a system with a small communication range this framework can still improve the system efficiency.

Concisely, the problem can be defined as a set of AUTOSAR runnables (*N* runnables) which have to be grouped into a set of tasks. The number of tasks is one the outputs of the framework which could be a number between one and *N*. The created task set should then be allocated among processing cores. A sequence of runnables creates a transaction. There is an end-to-end latency associated to each transaction. Each transaction is corresponding to a mission in the system, for example a brake system in a car. Therefore, the system can be described as a set of transactions. In order to fulfill the mission of a transaction, the sequence of runnables belong to this transaction should be executed in the predefined order to satisfy the data dependency between them. Accordingly, not only the framework should minimize the inter-runnable communication cost but it should also be able to meet all of the end-to-end latencies of transactions.

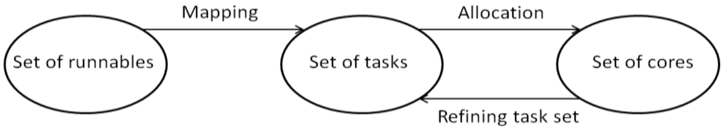


Figure 2: Framework for mapping of AUTOSAR runnables to OS tasks and allocation to multicores.

**Summary of Deliverables**

- Proposing two different solution frameworks to address the problem. In both frameworks both the process of mapping and allocation were considered together to reach a highly efficient solution (see Figure 2). In addition, in both frameworks a refining function is defined in which after finding an allocation of tasks to cores, the initial created task set is refined to minimize the number of tasks. In other words, this function attempts to merge the tasks located on the same core as much as possible to reduce the communication cost.
- The core part of both mentioned solution frameworks are some heuristics algorithms which were introduced to cope with the problem within a reasonable execution time. As the nature of the problem is NP-hard, an exhaustive search does not allow us to find a solution within a reasonable execution time. Therefore heuristic approaches were considered.
- In one of the frameworks the architecture of the multi-core processors on which a set of AUTOSAR runnables have to be mapped is considered with more details. We have suggested a communication cost model for a multi-core processor with three levels of cache memories. The model can easily be adapted for a two level cache processor.
- We also considered the problem when we have a set of transactions while some of them share some runnables (called dependent transactions) by this assumption that runnables do not have any internal states.
- In the last step, we are solving the problem when we have dependent transactions while the runnables may have some internal states and thus, different instances of a runnable should not be executed by more than one transaction at the same time.

## 5.3 Results and deliverables related to challenge *performance*

Embedded software systems based on AUTOSAR are large and complex and many different tools are involved in the development. Analysis and synthesis of these systems require information at different abstraction layers and types. For example, information about software (e.g. software components and runnables, as defined in the AUTOSAR standard) and hardware (e.g. ECUs and networks) are important to systematically allocate software on ECUs and multicores. Information about how electronic components are connected and their characteristics are important to be able to systematically evaluate future system architecture alternatives.

Traditionally, automotive embedded systems are built up using a large number of ECUs with hardware specifically tailored to their particular function. Some of the drawbacks of this type of structure come from the fact that it is very heterogeneous, where basically each ECU is tailor-made for its specific function.

Some drawbacks are inefficient use of computational power, slow evolution of functionality, complex maintenance and large amount of unique hardware items. There are indications that the current E/E system architectures are more complex that they need to be to support the implemented functionality.

Virtual platform technologies have the potential to enable a more efficient embedded systems development process. A virtual platform is an embedded computer system simulated on a regular PC. It behaves like the physical target hardware and can boot and run the complete software stack. Virtual platforms have many advantages across the product life cycle, e.g. they enable massive parallel testing on clusters of PCs, thereby shifting work to a virtual environment, thereby improving development efficiency.

Programmable processors or System-on-Chips (SoCs) are potential alternatives to the conventional processors mainly used in automotive industry today. They consist of a fixed and a programmable logic part (i.e. an FPGA). The fixed part includes single or multi-cores and the programmable part can be tailor-made by including extra IP blocks, e.g. additional CAN controllers. The SoCs provide high performance and flexibility (one one may even envision that they will be programmed in production in the future), but they also require changes to the current development processes because the hardware and software of SoCs are developed concurrently (i.e. hardware/software co-design).

To evaluate the impact of different hardware architectures – single- and multicore microcontroller architectures - as well as software architecture utilizing a multicore environment, a demonstrator is compiled. The purpose of the Multicore Demonstrator is to show the performance difference between a single core platform running an application (or more) and a suite of AUTOSAR modules, and a multi core platform running the same application(s) but with some of the AUTOSAR basic software modules offloaded to another core. In particular, the AUTOSAR basic software modules that will be offloaded to another core in the demonstrator are those belonging to the communication stack.

For the demonstrator, a board is used which offers an asymmetrical dual-core implementation, with the slave core being considered more of an I/O processor.

**Summary of Deliverables**

- Development of software tools for analysis of complex embedded AUTOSAR systems at different abstraction layers and types [pub8]. Examples of information analyzed are AUTOSAR specific XML files, end-to-end timing requirements, CAN and LIN database files, wiring diagram and characteristics of ECUs in terms of analog and digital inputs and outputs pins.
- Literature study of integrated system architectures that potential may be used in automotive industry to address identified short-comings of the current system architectures [pub9]. It was found that the automotive industry may potentially learn from concepts that have been developed in other industries (e.g. avionics) to address the same type of challenges that now is facing the automotive industry.
- Literature study of virtual platform technology in general, a market survey of commercial virtual platforms and tools, and an evaluation of open source software to better understand advantages and disadvantages of the technology [pub10]. In addition, work has been carried out together with one supplier to tailor-made their virtual platform software for real-use in the automotive industry.
- Evaluation of different model-based workflows for hardware/software co-design based on Matlab/Simulink models. A suitable workflow was found which was then applied to an advanced driving assistance application to get further knowledge of how SoCs may be used in future automotive products [pub15].
- A demonstrator based on an asymmetrical dual-core architecture running an AUTOSAR platform. A base case implementation – a single-core scenario which then can used as a base for comparing different multi-core scenarios
- Implementations of different multi-core scenario based on AUTOSAR multi-core concepts, evaluation of benefits and drawbacks for the different scenarios, performance comparisons.

- Conclusions based on the findings from the demonstrator and the evaluated scenarios, initial basic guide lines of potential ways of selecting scenarios for different use cases **Error! Reference source not found.**.

### 5.4 Delivery to FFI-goals

Table 1 shows estimates of how the project contributes to the targets set forth in the program FFI Fordonsutveckling, version 2011-02-01. The original text is in Swedish has been translated by Volvo Technology.

| **Vision**<br>*Swedish text is taken from the programme description, version 2011-02-01. English translation by Volvo Technology.* | **Level**<br>(Low, Medium, High) |
|---|---|
| **Specific for FFI Vehicle Development** | |
| *The Swedish automotive industry is at the forefront with vehicles, vehicle components and development services that are safe, environmentally friendly and energy efficient.* | High |
| **E/E Systems/Embedded Systems and Software** | |
| *Establish national competence that is able to develop complex embedded software systems. Green, Safe and Connected vehicles require high national skills which are capable of developing complex electrical systems that use both a national and a global range of research and technology.* | High |
| **Materials Technology for more Efficient Vehicles** | |
| *The automotive industry has got useful and innovative materials and access to innovative use of materials.* | |
| *Substantial (measurable) weight reduction* | Meduim |
| *Substantial cost reduction* | Medium |
| *Significantly better material properties* | Low |
| **Methods and Tools for Vehicle Development** | |
| *Establish world-leading methodologies and tools for vehicle development.* | High |

Table 1: Summary of visions taken from the FFI program description, version 2011-02-01.

## 6. Dissemination and publications

A number of dissemination activities have been performed at the project participant's premises and publications have been presented at international conferences (see below).

**Papers**

[pub1] Evaluating Industrial Applicability of Virtualization on a Distributed Multicore Platform, Nesredin Mahmud, Kristian Sandström, Aneta Vulgarakis, The 19th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2014).

[pub2] A Communication-Aware Solution Framework for Mapping AUTOSAR Runnables on Multi-core Systems (Sep 2014). Hamid Reza Faragardi, Kristian Sandström, Björn Lisper, Thomas Nolte. The 19th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2014)

[pub3] An Efficient Scheduling of AUTOSAR Runnables to Minimize Communication Cost in Multi-core Systems (Sep 2014). Hamid Reza Faragardi, Kristian Sandström, Björn Lisper, Thomas Nolte. The Seventh International Symposium on Telecommunications (IST2014)

[pub4] Towards a communication-aware mapping of software components in multi-core embedded real-time systems (Apr 2014). Hamid Reza Faragardi, Kristian Sandström, Björn Lisper, Thomas Nolte. The 20th IEEE Real-Time and Embedded Technology and Applications Symposium, WiP session (RTAS 2014 WIP)

[pub5] Communication-aware Scheduling of AUTOSAR Runnables on Multi-core Systems (Apr 2014). Hamid Reza Faragardi, Kristian Sandström, Björn Lisper, Thomas Nolte. International Workshop on Design Space Exploration of Cyber-physical Systems (IDEAL 2014)

**Reports**

[pub6] 9ADB005953-009 Autosar_Industrial_Automation. Aneta Vulgarakis, Markus Lindgren, Kristian Sandström. 2013.

[pub7] 9ADB005953-012 Partitioned models. Aneta Vulgarakis, Markus Lindgren, Kristian Sandström. 2013.

[pub8] Analysis of Complex AUTOSAR Systems. Anders Svensson. 2014.

[pub9] Integrated System Architectures. Anders Svensson. 2014.

[pub10] Virtual Prototyping for Embedded Software. Mats Olsson. 2014.

[pub11] Multicore project - final report. ArcCore. 2014.

**Theses**

[pub12] A Frame Packing Framework for Control Systems. Sylvain Gabry. Master Thesis, KTH/ICT. 2014.

[pub13] Automated Orchestra for Industrial Automation on Virtualized Multicore Environment. Nesredin Mahmud. Master Thesis, Mälardalen högskola/Akademin för innovation, design och teknik. 2013.

[pub14] Resource based analysis of Ethernet communication between software partitions. Cezar Chiru. Master Thesis, Mälardalen högskola/Akademin för innovation, design och teknik. In review.

[pub15] A Driving Assistance System with Hardware Acceleration. Gongpei Cui. Master Thesis, Chalmers University of Technology (Computer Systems and Networks program). In review

# 7. Conclusions and future research

This project, as a whole, aims to achieve resource efficient and predictable usage of multi-core technology in the context of AUTOSAR. The AUTOSAR standard has as a goal to standardize substantial aspects of the software development for automotive systems. Even though AUTOSAR has been developed with automotive applications in focus, the AUTOSAR software architecture shares many elements with those that can be found in the industrial automation domain. It has therefore been a common reference point shared between the different stakeholders in the project. As such the transfers of results between different stakeholders have been successful.

The multicore technology provides several challenges and opportunities that to a large extent are shared among different industrial domains. The project has delivered valuable results for the participating companies in the areas of integration, performance, and predictability of future multicore-based systems.

Future research may investigate how multicore technology can improve future system E/E architectures. Examples of drawbacks in current architectures are inefficient use of computational power (some ECUs will be pushed to their limits in computational power, while other ECUs use less than their capacity), slow evolution of functionality (with each ECU being tailor-made, all software for the ECU needs to be tailor-made), complex maintenance (the more heterogeneous the system is the more complex the maintenance, as every ECU has its own failure characteristics) and high amount of unique hardware items (a large number of unique hardware items brings with it issues like managing a large amount of part numbers, inventory, after market considerations).

# 8. Participating parties and contact person

ABB AB
Kristian Sandström
kristian.sandstrom@se.abb.com

ArcCore AB
Michael Svenstam
michael.svenstam@arccore.com

Mälardalen University
Björn Lisper
bjorn.lisper@mdh.se

Mälardalen University
Thomas Nolte
thomas.nolte@mdh.se

Volvo Technology AB
Anders Svensson
anders.svensson.4@volvo.com

# 9. References

[1] K. Asanovic et al. The Landscape of Parallel Computing Research: A View from Berkeley, Technical Report, EECS Department, University of California, Berkeley, Dec. 2006.
[2] AUTOSAR – AUTomotive Open System ARchitecture, http://www.AUTOSAR.org/
[3] Z. Deng and J. W.-S. Liu. Scheduling real-time applications in an open environment. In 18th IEEE Real-Time Systems Symposium (RTSS'97), pp. 308–319, Dec. 1997.
[4] T.-W. Kuo and C. Li. A fixed-priority-driven open environment for real-time applications. In 20th IEEE Real-Time Systems Symposium (RTSS'99), pp. 256–267, Dec. 1999.
[5] G. Lipari, J. Carpenter, and S. Baruah. A framework for achieving inter-application isolation in multiprogrammed hard-real-time environments. In 21st IEEE Real-Time Systems Symposium (RTSS'00), pp. 217–226, Dec. 2000.
[6] I. Shin and I. Lee. Periodic resource model for compositional real-time guarantees. In 24th IEEE Real-Time Systems Symposium (RTSS'03), pp. 2–13, Dec. 2003.
[7] I. Shin and I. Lee. Compositional real-time scheduling framework. In 25th IEEE Real-Time Systems Symposium (RTSS'04), pp. 57–67, Dec. 2004.
[8] R. I. Davis and A. Burns. Hierarchical fixed priority pre-emptive scheduling. In 26th IEEE Real-Time Systems Symposium (RTSS'05), pp. 389–398, Dec. 2005.
[9] G. Lipari and E. Bini. Resource partitioning among real-time applications. In 15th Euromicro Conference on Real-Time Systems (ECRTS'03), pp. 151–158, Jul. 2003.
[10] S. Saewong, R. Rajkumar, J. P. Lehoczky, and M. H. Klein. Analysis of hierarchical fixed-priority scheduling. In 14th Euromicro Conference on Real-Time Systems (ECRTS'02), pp. 152–160, Jun. 2002.
[11] I. Shin, A. Easwaran, and I. Lee. Hierarchical Scheduling Framework for Virtual Clustering of Multiprocessors. In 20th Euromicro Conference on Real-Time Systems (ECRTS'08), pp. 181-190, Jul. 2008.
[12] G. Lipari, and E. Bini. A Framework for Hierarchical Scheduling on Multiprocessors: From Application Requirements to Run-Time Allocation. In RTSS 2010, pp. 249-258, Dec. 2010.
[13] E. Bini, G. C. Buttazzo, and M. Bertogna. The Multi Supply Function Abstraction for Multiprocessors. In RTCSA 2009, pp. 294-302, Aug. 2009.
[14] T. Baker. Stack-based scheduling of real-time processes. Journal of Real-Time Systems, 3(1):67–99, 1991.
[15] J. Carpenter, S. Funk, P. Holman, A. Srinivasan, J. Anderson, and S. Baruah. A categorization of real-time multiprocessor scheduling problems and algorithms. In Handbook on Scheduling Algorithms, Methods, and Models. Chapman Hall/CRC, Boca, 2004.
[16] U. Devi. Soft real-time scheduling on multiprocessors. PhD thesis, available at www.cs.unc.edu/˜anderson/diss/devidiss.pdf, 2006.
[17] AUTOSAR. Specification of Multi-Core OS Architecture (Ver. 1.1.0), Nov. 2010.
[18] I. I. Lupu, P. Courbin, L. George, and J. Goossens. Multi-criteria evaluation of partitioning schemes for real-time systems. In ETFA 2010, pp. 1-8, Sep. 2010.
[19] C. J. Kenna, J. L. Herman, B. B. Brandenburg, A. F. Mills, and J. H. Anderson. Soft Real-Time on Multiprocessors: Are Analysis-Based Schedulers Really Worth It? In RTSS 2011, pp. 93-103, Dec. 2011.

[20] B. B. Brandenburg, J. M. Calandrino, A. Block, H. Leontyev, and J. H. Anderson. Real-Time Synchronization on Multiprocessors: To Block or Not to Block, to Suspend or Spin? In IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'08), pp. 342-353, Apr. 2008.
[21] P.-E. Hladik, A.-M Deplanche, S. Faucou, and Y. Trinquet. Adequacy between AUTOSAR OS specification and real-time scheduling theory. In IEEE International Symposium on Industrial Embedded Systems (SIES'07), pp.225-233, Jul. 2007.
[22] N. Navet, A. Monot, B. Bavoux, and F. Simonot-Lion. Multi-source and multicore automotive ECUs - OS protection mechanisms and scheduling. In IEEE International Symposium on Industrial Electronics (ISIE'10), pp.3734-3741, Jul. 2010.
[23] A. Monot, N. Navet, B. Bavoux, and F. Simonot-Lion, Multicore scheduling in automotive ECUs. In Embedded Real Time Software and Systems (ERTSS'2010), May 2010.
[24] S. Anssi, S. Tucci-Piergiovanni, S. Kuntz, S. Gerard, and F. Terrier. Enabling Scheduling Analysis for AUTOSAR Systems. In 14th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC'11), pp.152-159, Mar. 2011.
[25] MAST (Modeling and Analysis Suite for Real-Time Applications) software tool, http://mast.unican.es/
[26] J. Rosen, A. Andrei, P. Eles, and Z. Peng. Bus Access Optimization for Predictable Implementation of Real-Time Applications on Multiprocessor Systems-on-Chip. In 28th IEEE International Real-Time Systems Symposium (RTSS'07), pp. 49-60, Dec. 2007.
[27] S. Chattopdhyay, C.-L. Kee, A. Roychoudhury, T. Kelter, P. Marwedel, and H. Falk. A Unified WCET Analysis Framework for Multi-core Platforms. To appear in 18th IEEE Real-time and Embedded Technology and Applications Symposium (RTAS'12), Apr. 2012.
[28] Y. Li, V. Suhendra, Y. Liang, T. Mitra, and A. Roychoudhury. Timing Analysis of Concurrent Programs Running on Shared Cache Multi-Cores. In 30th IEEE International Real-Time Systems Symposium (RTSS'09), Dec. 2009.
[29] N. Böhm et al., A Comparison of Pragmatic Multi-Core Adaptations of the AUTOSAR System. In OSPERT'11, Jul. 2011.
[30] Amdahl's law, http://en.wikipedia.org/wiki/Amdahl's_law
[31] MC2-AI: Multi-Core Computing in Automotive Industry, FFI –project, http://www.vinnova.se/sv/Resultat/Projekt/Effekta/MCC-AI-Multi-Core-Computing-in-Automotive-Industry/
[32] Integrated Modular Avionics (IMA), http://en.wikipedia.org/wiki/Integrated_modular_avionics
[33] A. Sangiovanni-Vincentelli, and M. Di Natale. Embedded System Design for Automotive Applications. Vol. 40, Issue. 10, pp. 42 - 51, Oct. 2007.
[34] A. Sangiovanni-Vincentelli. Moving From Federated to Integrated Architectures in Automotive: The Role of Standards, Methods and Tools. In Proceedings of the IEEE, Vol. 98, Issue. 4, pp. 603 - 620, Apr. 2010.
[35] W. Zheng et al. Synthesis of Task and Message Activation Models in Real-Time Distributed Automotive Systems. In IEEE/ACM DATE Conf. IEEE Press, 2007, pp. 93-98.
[36] A. Davere et al. Period Optimization for Hard Real-Time Distributed Automotive Systems. In Proc. ACM Design Automation Conf., ACM Press, 2007, pp. 278-283.
[37] F. Balarin et al. Metropolis: An Integrated Electronic System Design Environment, Computer, Vol. 36, Issue: 4, Page(s): 45 - 52, 2003.
[38] ATESST2 project, http://www.atesst.org/scripts/home/publigen/content/templates/show.asp?L=EN&P=111
[39] TIMMO project, http://www.itea2.org/project/index/view/?project=170
[40] Tilera Corporation, Official web-site, http://www.tilera.com