Seamless Model-Based Safety Engineering from Requirement to Implementation

Georg Macher

Institute for Technical Informatics
Graz University of Technology
AUSTRIA

Abstract. Development of embedded automotive systems has become tremendously complex in recent years. The trend of replacing traditional mechanical systems with modern embedded systems enables deployment of more advanced control strategies. This provides new benefits for the customer and environment, but at the same time, the higher degree of integration and safety-criticality raise new challenges. In parallel new automotive safety standards, such as ISO 26262, and the introduction of automotive multi-core systems require efficient and consistent product development. To tackle the issues of mixed-critical multi-core systems development with hard real-time constraints and provide academical methodologies and approaches the MEMCONS project was launched. Aim of this paper is to provide an overview of the scientific research problem, approaches to solve the problem and ways to evaluate the solution found by the project related PhD thesis.

1 Problem Statement

Embedded electronic control systems are strong innovation drivers for the automotive industry. The number of embedded systems has significantly grown in recent years and novel multi-core computing platforms are even stronger innovation drivers. This technology enables more advanced control strategies and increase the degree of integration and complexity of such systems. Nevertheless, safety-critical system development according industry standard ISO 26262 [6] has to be ensured.

The issues appearing in this context are manifold. Safety-critical system development according automotive standards requires safety conception along the whole development process, starting from initial development to final decommissioning of the product. Safety is a system-wide, cross-domain feature which needs to be considered in each development step by each involved department. Therefore the classical 'divide & conquer' approach of the automotive domain has to be reconsidered.

Secondly, the automotive safety standard introduces additional development artifacts, constraints, and a standardized development process. However, automotive related organizations already have their own (safety) processes in place, certified and process-skilled employees, therefore they are unwilling or unable to

migrate their process quickly. Instead, the required safety activities need to be integrated within the existing process and tool landscape.

As third, a conceptual change from document-centric development approaches to model-based development (MBD) approaches needs to be forced to ensure required traceability, maintainability, reuse, and certifiability of development decisions and products [2].

The second main focus, multi-core systems, includes equivalent open issues. Currently methodologies and tools supporting safety-critical development of multi-core systems are yet hardly available. Also industry standards, like AUTOSAR [1], are currently not covering the multi-core related challenges. Main rising challenges in this context, beside traceability issues, are parallelization of state-of-the-art software architectures and tracing of dependencies and bottlenecks of multi-core systems. Side-effects and unintended correlations need to be traced and tackled with adequate methodologies to ensure freedom from interference for safety-critical applications.

Figure 1 illustrates the identified open issues and relevant problem domain of this PhD thesis.

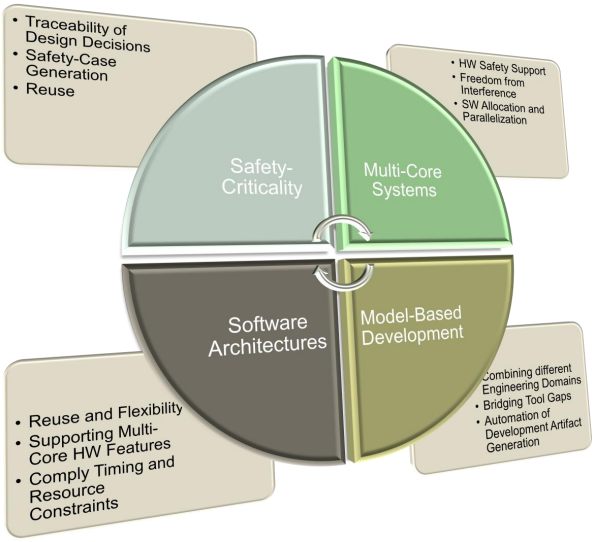

Fig. 1. Illustration of the Problem and Open Issues

We propose to extend the existing model-based development approach (further details see [8]) with a model representation of the hardware in use, an AUTOSAR aligned model for software development, and a hardware-software interface according to ISO 26262. Furthermore, this tool-chain is enhanced by extractors automatically generating system and electronic control unit (ECU) configuration files from existing information at system development level. This proposed

approach closes the gap, also mentioned by Giese et al. [3] and Holtmann et al. [5], between system-level development at abstract UML-like representations and software-level development modeling tools (e.g. Matlab Simulink/Targetlink). Closing this gap creates a seamless tool-chain from initial requirements (coming from a requirement management tool), through definition of safety concepts and software architectures (in a model-based development environment), to final decisions in code implementation in compliance with automotive safety standards.

2 Related Work

The related works for this thesis is manifold. Therefore, this section solely focus on related funding projects. Other related publications tackle solely parts of the project aims and have therefore been omitted due to page limitations.

The SAFE project[1] objective is to enhance methods for defining safety goals and define development processes complying with the new ISO26262 standard for functional safety in automotive electrical and electronic systems. Different to this project we are not focusing on collaboration of automotive companies. Furthermore, the focus of this project is put on extending AUTOSAR architectural models for supporting ISO 26262 product development at concept phase (part 3 of ISO 26262). In contrast to this, we focus on part 4 and 6 of the ISO 26262 norm (system- and software development).

The AMALTHEA project[2] focus is on development of an open source development platform with common data models and interfaces. Therefore, the focus of a common data model for safety critical system development and interfaces for supporting the data exchange between development tools is similar to ours, but we also consider automatic checking for safety-related and multi-core related constrains (such as execution order effects on timings and supporting ASIL decomposition features). In addition, we also intend to assemble a collection of patterns to be applied for safety-critical multi-core system development.

The Model-based analysis and engineering of novel architectures for dependable electric vehicles (MAENAD) project [3] focuses similar topics but in relation to pure electric vehicles and based on EAST-ADL2. In difference to this project, we also focus on automated techniques for constraint checking of multi-core features and automated transfer of information between special purpose software tools (such as RTOS configurators or RTE generators).

The project SPES_XT[4] also focus on methodology and integration of development tools within a seamless tool-chain. Other than this project, we solely focus on the automotive domain, therefore we aim to achieve a methodology more specialized for the needs of the automotive domain, but in contrast to the SPES_XT project, we deal with the topics of safety-criticality and multi-core systems more detailed than this project.

[1] http://safe-project.eu/
[2] http://amalthea-project.org/
[3] http://www.maenad.eu/
[4] http://spes2020.informatik.tu-muenchen.de/spes_xt-home.html

The CESAR project [12] proposes cost-efficient methods and processes for the development of safety relevant embedded systems. Integrated tool chains are moving the engineering disciplines together and provide traceability along the development process. Main focus of the proposed tool chains in CESAR are related to systems and safety engineering. The introduced multi-domain approach, European cross-sectoral standard reference technology platform (RTP), provides meta-models and methods. But for less abstract development phases the RTP needs to be more specific and refined to tighter couple inter-operations between different tools.

3 Proposed Solution

The approach relies on automotive system model representation and tool bridges based on domain standard exchange formats (such as AUTOSAR XML [1] or OSEK/VDS OIL files [11]). Therefore it is possible to import existing AUTOSAR components, interface configuration and timing constraints (AUTOSAR *R*4.0) into the system model. Figure 2 shows the conceptual overview of the approach, and highlights the bridging approach on tool level. As can be seen in this figure several independent tools are linked via specific interfaces (highlighted in yellow) to a seamless development tool chain, using the system model representation a common source of information.

Furthermore, the automatic export of component containers and their interconnections is possible, which links the software architecture designed in SysML to the software development tool (e.g. Matlab/Simulink) and closes the gap between system development tools and functional software development tools. We also take into account automotive constraints (especially traceability requirements) and close the existing tool gap between basic software configuration tools, operating systems (OS), and scheduling tools.

Automotive OS do not have dynamic scheduling parts, therefore all OS settings are static and can be specified during the development phase. The available information from system development can be exported and used to integrate OS and scheduling tools to automatically generate a distribution of tasks onto cores. Our approach therefore also helps to specify tasks with their priority, duration, and safety-criticality, the mapping of tasks to cores, generate task activation policies, and support specification of task resources, alarms, and interrupts.

An additional advantage for multi-core systems is based on the definition of the software architecture in our system development environment and the automatic configuration of safety drivers, BSW, and RTE, which can be generated from the SysML representation. Within this environment the allocation of software components to cores can be changed and supported more easily via automatic approaches, e.g. collection of safety-relevant software on one specific core or a switch to static work balancing between cores. In addition, tasks, inter-core communications, and synchronizations can be investigated at this higher abstraction level, and resource bottlenecks can be minimized earlier. Furthermore, different compilers, linkers, and even development or configuration tools can be

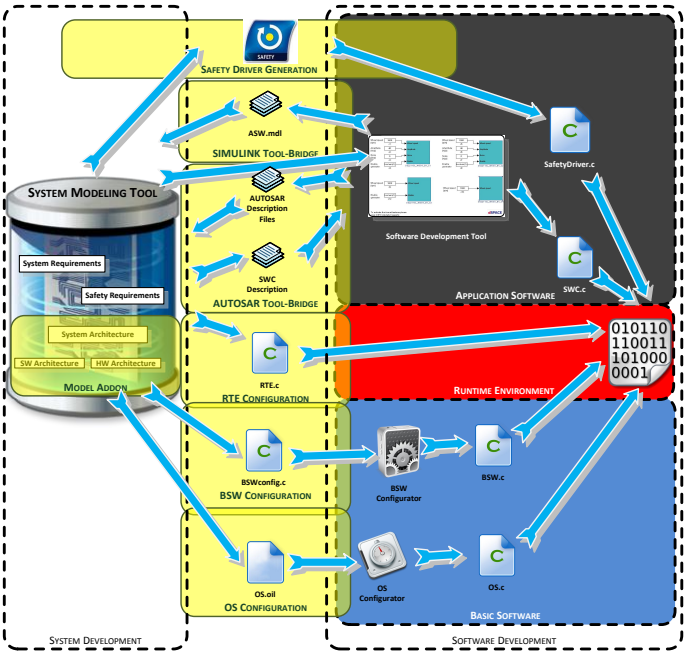

Fig. 2. Illustration of the Proposed Solution in Tool Level Viewpoint

used within the established tool-chain due to its linkage via domain interchange formats. This allows the inclusion of additional multi-core configuration tools or task distribution tools into the tool-chain (e.g. analysis tool presented by Hilbrich et al.[4]).

The contribution composed of tool add-ons and their respective base methodology is given in Table 1.

4 Preliminary Work

The work status of to-date for the tool integration is also mentioned in Table 1. The status of the basic methodologies for the bridging approach and traceability solution is ahead of the numbers mentioned for the tool implementation in Table 1. The methodology for integrating all required information into the model-based system development database is done, except of minor changes.

The integration of application software departments is based on the definition of the SW architecture within the system development tool. This information is then transferred either via the AUTOSAR aligned exchange methodology (based on ARXML files) or via API directly to the special purpose tools of application software developers.

The generation of dedicated HW safety feature drivers and the mapping between software modules (ASW to ASW as well as ASW to BSW) is done

directly via .c and .h files. The methodology for BSW configuration is also based on direct insert of C structs within .c and .h files. For the OS configuration the methodology bases on the OSEK/VDX OIL approach, which is also a domain standard.

Other tool bridgings, such as the test environment or the requirement management tool, also need to rely on domain standard exchange formats or are required to implement an API to interchange with the system modeling tool.

Table 1. Approach Improvement Indicators

Tool addon	**short description**	**Illustration**	**work status**
UML modeling framework	for modeling of HW and SW of multi-core systems	Figure 2 - Model Addon	90 %
OSEK/ VDX OIL import/ exporter	to generate OS configurations	Figure 2 - OS Configuration	100 %
BSW configurator	for configuration of basic software modules	Figure 2 - Model BSW Configuration	10 %
RTE configurator	to generate links between BSW and ASW automatically	Figure 2 - RTE Configuration	0 %
AUTOSAR import/ exporter	for application software module description interchange	Figure 2 - AUTOSAR Tool-Bridge	85 %
SIMULINK import/ exporter	for non-AUTOSAR ASW module description interchange	Figure 2 - SIMULINK Tool-Bridge	25 %
Safety driver generator	for specific safety HW configuration	Figure 2 - Safety Driver Generation	0 %
Test tool bridge	for integration of test environment	not illustrated in a figure	10 %
HW - SW interface definition import/ exporter	exchange of HW and SW interface definitions	Figure 2 - part of the Model Addon	100 %

5 Expected Contributions

The contributions of this PhD thesis are on one hand to provide method descriptions and tool prototypes to integrate the required automotive safety activities within the existing process and tool landscape of our industrial project partner. On the other hand, we aim to provide a pattern catalog as guidance for safety-critical system development with multi-core systems and an use-case example for training purpose. As a third we aim to improve or define (if not available) basic methodologies for multi-core system development and parallelization of

state-of-the-art software architectures, tracing of dependencies and side-effects. Parts of the tool-chain have already been published:

- Bridging Automotive Systems, Safety and Software Engineering by a Seamless Tool Chain, ERTS2014, Feb 2014 [8]
- Automated Generation of AUTOSAR Description File for Safety-Critical Software Architectures, Informatik2014, Sept 2014 [7]
- Automated Synchronization of System Architecture and Automotive Real-time Operating Systems, Embedded Operating Systems, Nov 2014 [9]

An initial approach towards collection of pattern for automotive safety-related system development has also been published:

- Pattern-Based Automotive Safety Cases: An Industrial Case Study, EuroPloP, July(Dec) 2014 [10]
- SAHARA - A Security-Aware Hazard and Risk Analysis Method, DATE Conference, Mar 2015, currently pending

6 Plan for Evaluation and Validation

Evaluation for the contribution has to be done in several steps, because of the varying contribution levels. These measures will be compared to the numbers of previously available tool-chains and methods based on an automotive use-case. To evaluate the tool prototypes the following performance indicators can be investigated:

- number of generated configurations
- number of additional information transferred between tools
- number of information lost by forward and backward model update
- speedup in time
- number of automatically generated documentations

Performance indicators for the evaluation of the methodologies are:

- number of automatic generated artifact traces
- number of traceable relations between initial requirement and final implementation
- number of automatic constraint checks
- useability evaluation with use-case
- relevance feedback of pattern from engineers
- impact analysis of supported information from engineers
- speedup in training time
- acceptance of publications at domain specific conferences

7 Current Status

This section concludes the paper with an overview of the current project status and progress. A rough overview was already given in Section 4 and can be seen for the tool implementations in Table 1.

The approaches for software model transfer based on AUTOSAR files (published in [7]) is currently under rework and evaluation by master student thesis.

The OS configuration tool bridging is also currently in evaluation at our industrial project partner (published in [9], depicted in Figure 2 - OS Configuration). Furthermore, the software model transfer via tool API (see Figure 2 - SIMULINK Tool-Bridge), BSW configurator (see Figure 2 - Model BSW Configuration), and RTE configurator (Figure 2 - RTE Configuration) are currently in development together with student with automotive background. The integration of the test environment of our industrial partner is in coordinated development together with the partner. Other open points will be addressed this years fall. An first test run of the whole tool prototypes is expected for begin of next year. The first quarter of the upcoming year is then planed for further improvement of the approach. The whole thesis is intended to be finished till next years fall.

References

1. AUTOSAR development cooperation. AUTOSAR AUTomotive Open System ARchitecture, 2009.
2. Manfred Broy, Martin Feilkas, Markus Herrmannsdoerfer, Stefano Merenda, and Daniel Ratiu. Seamless Model-based Development: from Isolated Tool to Integrated Model Engineering Environments. *IEEE Magazin*, 2008.
3. Holger Giese, Stephan Hildebrandt, and Stefan Neumann. Model Synchronization at Work: Keeping SysML and AUTOSAR Models Consistent. *LNCS 5765*, pages pp. 555 –579, 2010.
4. Robert Hilbrich and Hans-Joachim Goltz. Model-based Generation of Static Schedules for Safety Critical Multi-Core Systems in the Avionics Domain. In *WMSE11*, 2011.
5. Joerg Holtmann, Jan Meyer, and Matthias Meyer. A Seamless Model-Based Development Process for Automotive Systems, 2011.
6. ISO - International Organization for Standardization. ISO 26262 Road vehicles Functional Safety Part 1-10, 2011.
7. Georg Macher, Eric Armengaud, and Christian Kreiner. Automated Generation of AUTOSAR Description File for Safety-Critical Software Architectures. In *Lecture Notes in Informatics*, 2014.
8. Georg Macher, Eric Armengaud, and Christian Kreiner. Bridging Automotive Systems, Safety and Software Engineering by a Seamless Tool Chain. In *7th European Congress Embedded Real Time Software and Systems Proceedings*, pages 256 –263, 2014.
9. Georg Macher, Muesluem Atas, Eric Armengaud, and Christian Kreiner. Automotive Real-time Operating Systems: A Model-Based Configuration Approach. In *ACM SIGBED Review Special Interest Group on Embedded Systems*. Association for Computing Machinery. Special Interest Group on Embedded , 2014.
10. Georg Macher, Harald Sporer, and Christian Kreiner. Pattern-Based Automotive Safety Cases: An Industrial Case Study. In *Conference Proceedings EuroPloP2014*, 2014.
11. OSEK/VDX Steering Committee. OSEK/VDX System Generation OIL: OSEK Implementation Language. http://portal.osek-vdx.org/files/pdf/specs/oil25.pdf, 2004.
12. Ajitha Rajan and Thomas Wahl. *CESAR Project Book*. Springer Verlag, 2012.