Defining a Timing Model for AUTOSAR – Status and Challenges

Kai Richter [1]

CTO, Symtavision GmbH
Frankfurter Straße 3B
38122 Braunschweig
richter@symtavision.com

Abstract: Software timing aspects have not received broad attention in the automotive industry until recently. New design trends and the ongoing work in AUTOSAR have significantly increased the industry's awareness to these issues. Now, timing is recognized a major challenge and has been put explicitly on the AUTOSAR agenda. The goal is to add timing models to the existing AUTOSAR templates. But it isn't all that simple. The paper highlights key technical and non-technical challenges for defining a comprehensive timing model for AUTOSAR, and outlines requirements for possible solutions. Examples from practice and a look into the industrial process of designing –and the way of thinking– shall help structuring the discussions. Finally, recent advancements of the new AUTOSAR timing subgroup and related projects are summarized.

1 Introduction

Current supply-chains in automotive E/E development contain hundreds of companies that design their individual hardware and software components based on requirement definitions from the OEMs or Tier-1 suppliers. Clearly, systems integration has become a key challenge.

To ease the integration in the future, the AUTOSAR partnership [He06] (http://www.autosar.org), an alliance of OEM manufacturers and Tier-1 automotive suppliers with many associates, has established a number of de-facto open industry standards for automotive E/E architectures. Many proven-valuable concepts are borrowed from earlier standards such as OSEK/VDX [OS04][OS05]. The main goal of AUTOSAR is to define a software architecture with standardized APIs and configuration files for application and basic software, which allows exchanging parts of the system's software in ways that programmers know from Java or C++. Key goals are modularity, scalability, transferability and re-usability of software among projects, variants, suppliers, customers, etc. without the time-consuming and costly need to re-configure, port, and re-build the code.

The standardized architecture is complemented with a standardized methodology that shall facilitate integrating components from different suppliers onto one ECU, re-mapping components from one ECU to another, or exchanging whole bus segments within a network of distributed functions; and (indirectly) new business models.

[1] This document is partially based on the TIMMO project in the framework of the ITEA 2, EUREKA cluster Programme Σ! 3674. The work of the German partners has been funded by the German Ministry for Education and Research (BMBF) under the funding IDs 01IS07002 (B-I,K). The responsibility for the content rests with the authors.

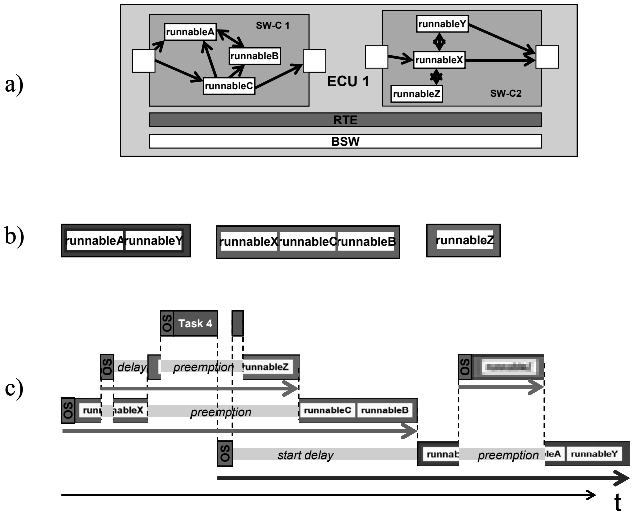

Figure 1 Software Components Structure vs. Task Structure

2 AUTOSAR and Timing

It is important to understand that the primary objective of AUTOSAR is not solving timing problems in particular but supporting integration from a software-engineering perspective. However, system timing properties have a strong impact on key steps in the newly envisioned AUTOSAR methodology. For instance, adding a software component to an existing ECU potentially introduces non-functional timing and performance interference with the original ECU software due to scheduling, arbitration, blocking, buffering etc., eventually generating hard-to-find timing problems, including transient overload, buffer under- and over-flows, and missed deadlines that can finally make the new or the old functions or the entire ECU fail.

Tier-1 suppliers must control such “timing side-effects” which requires knowing timing properties of more or less each involved component. This is already a challenge in the established design process in which a Tier-1 supplier typically has control over the entire software running on a particular ECU. With AUTOSAR, the Tier-1s have to deal with black-box SW components and must still be able to control timing. Of course, not having a systematic timing analysis procedure is challenging future design processes.

There is a large agreement that AUTOSAR needs a timing model. But it isn’t all that simple. In the next section we provide three examples of (technical) model mismatch between the software-engineering view of AUTOSAR and the implementation-concept view needed for timing analysis [Ri07a]. Methodology and supply-chain concerns add to that dilemma, which are summarized in Section 4. We outline recently started activities and their goals in Section 5, followed by a general conclusion.

3 Model Mismatch

In this section we present three key examples of model mismatches that emphasize the complex relations between the timing properties of the system components.

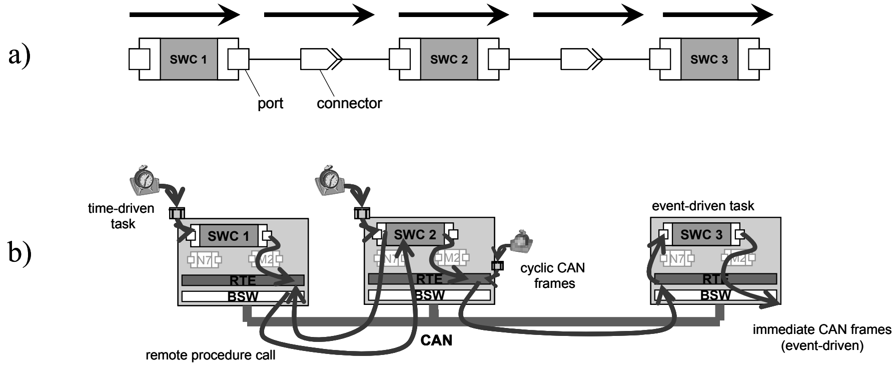

Figure 2 AUTOSAR VBF View and Implementation Mechanisms of "End-to-End Timing Chains"

Software Components and Tasks: At the VFB (virtual function bus) level, AUTOSAR defines so called *software components (SW-Cs)* as atomic entities (see Figure 1 a). However, when it comes to implementation, each SW-C comprises several so called *runnables* that are grouped into *tasks* (see Figure 1 b), which are finally put under operating system control. As a result from OS scheduling, runnables from any SW component can possibly interfere with any other runnable, irrespective of the actual SW-C structure (see Figure 1 c), finally leading to an implementation-dependent timing behavior crossing the SW component boundaries and introducing hidden timing dependencies between seemingly independent SW-Cs at the VFB level.

Logical End-to-End Timing and Interaction Mechanisms: With the increasing distribution of functions over several ECUs in a car, the importance of end-to-end timing (and deadlines) is also increasing. At the VFB level, AUTOSAR captures such *timing chains* composed of communicating *software components (SW-C)* that exchange *signals* via *connectors* and *ports* (see Figure 2 a). At the implementation level (see Figure 2 b), there exist several valid interaction and communication mechanisms including client-server (remote procedure call), periodic sampling including under- and over-sampling, polling, and event-driven. This leads to a variety of indirect *causality chains* that can significantly differ from the "logical causality". In absence of clear execution, communication, and buffering semantics, the timing will change with the implementation, uncontrolled by specification and untestable against test component models that lack the necessary details.

Complex Communication Drivers: A look at the communication driver structure reveals another type of mismatch. The AUTOSAR architecture defines a layered communication stack implementation that includes software and hardware buffers with different access strategies. Furthermore, AUTOSAR communication knows several *frame transmission modes (sporadic, periodic, and mixed)* and *signal transfer properties (triggered, pending)*. Despite their enormous influence on communication timing, these details are hardly visible at the AUTOSAR VFB level.

4 Methodology and Supply-Chain Issues

In addition to the formal definitions of a clear and intuitive timing model, designers also need a methodology to utilize that model in the established design flow. In order to deploy a reliable integration process, the supply-chain communications among all involved will have to evolve, most likely by establishing timing contracts between OEMs and suppliers. This means that not only a model for annotating timing properties to AUTOSAR elements is needed, but also mechanisms for distinguishing between a property, a guarantee, a requirement, a constraint, etc...

Furthermore, the scope of exchanged information is likely to change. Even though OEMs do not develop large parts of the software, they are responsible for the network in the center of integrations. The network timing, however, depends as well on protocol parameters such as CAN Id and FlexRay slot number as on ECU driver hardware and software (SW-Cs and COM stack), which is –today– mostly out of the OEM's control. In the future, such information must become part of the supply-chain communication. In turn, the Tier-1 suppliers have to cope with more and more software that is supplied externally as object code; examples include OEM-specific functions or basic software (BSW). The challenge is to design the ECU to meet all timing and performance requirements without knowing details of the supplied software [Ri07b].

Based on the feedback we –as a tool provider– are receiving from our customers, future supply-chain communications must have some key properties in order to be accepted:

- Responsibilities and scope must be clearly defined, and must match the established roles of suppliers and OEMs.
- IP protection must be ensured, in particular on the supplier's side. Together with already existing standards like AUTOSAR, this will have a dominant impact on the abstraction of a timing model.
- A comprehensive and reliable timing verification methodology must be in place, since there is no point in modeling something that cannot be analyzed.
- It must be clarified what kind of analysis results and what level of accuracy can be obtained, and the required effort. Full accuracy might not be needed, if only the results are significantly better than overly simplified spread sheets or „gut feeling".

While the OEMs and Tier-1s have certain very relevant requirements, the tool-suppliers (and academia) are the technology providers. We have successfully shown with several customers [He05, Je08] that timing modeling and analysis is possible with today's standards such as OSEK/VDX, and can be introduced in AUTOSAR. All parties together have a good chance to find reasonable timing modeling solutions for AUTOSAR but it is important to not expect the one and only solution. Timing challenges likely differ among domains. Control functions in the chassis domain are typical real-time applications where end-to-end deadlines can be found. Other domains such as infotainment might be more concerned with throughput and quality-of-service. Hence, there might be different types of models that suffice for one or the other domain.

5 Status and Recent Progress

The bad news is that neither the currently published AUTOSAR standard version 2.1 contains enough relevant aspects of timing and performance, nor will the next version 3.0 do. The good news is that two activities have started recently with significant industrial participation to tackle this situation. Since 2007, several OEMs and Tier-1s discuss timing modeling options with selected researchers and tool vendors in the EU-funded TIMMO project [Je07]. TIMMO's goal is defining a timing augmented description language (TADL) for software component models such as AUTOSAR, along with an appropriate methodology. Requirements are derived from scenarios that the industry partners (TADL users) deliver, while the technology partners contribute conceptual input. Bridging the gap between the API-centric software-engineering view and a math-centric timing analysis view is one key challenge, not only for the definition of the TADL but also to keep the discussion focused within TIMMO.

Meanwhile, also a new timing subgroup within AUTOSAR has been formed to introduce an initial set of core timing parameters into one of the next AUTOSAR releases [SR08], most likely focusing on the implementation view. Both activities tackle the challenges outlined in this paper but –due to their different timelines– with slightly different ambitions with respect to soundness, completeness, etc. AUTOSAR ends 2009, before TIMMO's results are finalized. The partners of both projects have committed to keep the work aligned.

6 Conclusion

AUTOSAR has introduced standardized APIs to ease system integration and enable new supplier-chain processes and business models (software as a product). However, not having a systematic timing modeling and analysis procedure in place is currently challenging key AUTOSAR goals. Several activities to add a "timing view" to the AUTOSAR standard have started recently, which is not all trivial. We have outlined several technical and non-technical challenges in the paper.

A sound solution must tackle modeling and analysis aspects as well as methodology and tools. Few tool vendors, typically those with an academic backing or background, already have parts of the solution available, and their participation in TIMMO and AUTOSAR is inevitable. First internal results indicate that the involved parties approach each other within a bounded scope of technical problems, clear goals, and willing to compromise because there will be no single solution that fits all. First results can be expected in AUTOSAR version 4.0.

7 References

[He05] R. Henia, A. Hamann, M. Jersak, R. Racu, K. Richter, R. Ernst. System Level Performance Analysis - the SymTA/S Approach. In IEE Proceedings Computers and Digital Techniques, Vol. 152, Is. 2, March 2005.

[He06] H. Heinecke, J. Bielefeld, K.-P. Schnelle, N. Maldener, H. Fennel, O. Weis, T. Weber, J. Ruh, L. Lundh, T. Sandén, P. Heitkämper, R. Rimkus, J. Leflour, A. Gilberg, U. Virnich, S. Voget, K. Nishikawa, K. Kajio, T. Scharnhorst, B. Kunkel. AUTOSAR—Current results and preparations for exploitation. In Proc. 7th EUROFORUM "Software in the vehicle". Stuttgart, Germany, May 2006

[Je07] M. Jersak et.al. Timing-Modell und Methodik für AUTOSAR. Elektronik automotive, Special issue on "AUTOSAR", October 2007

[Je08] M. Jersak. New kid on the block: Scheduling Analysis improves quality and reliability of ECUs, Busses and Networks. In Proc. Embedded World Conference, Nürnberg, Germany, February 2008.

[OS04] OSEK/VDX Communication. v.3.0.3, OSEK/VDX Consortium, July 2004

[OS05] OSEK/VDX Operating System. V.2.2.3, OSEK/VDX Consortium, February 2005

[Ri07a] K. Richter, On the Complexity of Adding Real-Time Properties to the AUTOSAR Software Component Model. In Proc. OMER4 Workshop. Paderborn, Germany, October 2007

[Ri07b] K. Richter. How OEMs can get Suppliers On Board for Designing Extensible Networks. In Proc. Embedded World Conference, Nürnberg, Germany, February 2007.

[SR08] O. Scheickl and M. Rudorfer. Automotive Real Time Development Using a Timing-augmented AUTOSAR Specification. In Proc. Embedded Real-Time Software Congress (ERTS), Toulouse, France, 2008