How Timing Interfaces in AUTOSAR can Improve Distributed Development of Real-Time Software

Oliver Scheickl, Michael Rudorfer, Christoph Ainhauser

BMW Car IT GmbH
Petuelring 116
80809 Munich, Germany
{Oliver.Scheickl | Michael.Rudorfer | Christoph.Ainhauser}@bmw-carit.de

Nico Feiertag, Kai Richter,

Symtavision GmbH
Frankfurter Straße 3B
38122 Braunschweig, Germany
{richter | feiertag }@symtavision.com

Abstract: Through the envisioned AUTOSAR methodology, the integration of hardware and software components shall become simpler and more flexible than ever. But in addition to standardized functional APIs, this approach also needs interfaces for non-functional component properties, specifically timing. However, formalizing the required parameters and defining new methodological steps is not trivial. In this paper, we illustrate ideas to structure timing properties by means of hand-over points and timing contracts using realistic examples, and we show how scheduling analysis provides key information to parameterize the resulting timing interfaces. Finally, we derive (sufficiently dark) gray-box timing chain segments that can be communicated between OEMs and suppliers without loss of IP protection. The resulting models are a) compatible with the AUTOSAR software architecture definitions and b) fit into the envisioned methodology.

1 Introduction

In today's cars, the number and complexity of electronic functions is ever increasing. Functions that were introduced as innovations in luxury class cars become standard functions a few years later. For example the current BMW 3 series has almost the same amount of functionality as the previous 7 series. (see Figure 1). At the same time, the hardware topology is evolving, as lifecycles in the semiconductor industry are significantly shorter than in the automotive industry. In this context, function interoperability and software re-use are necessary to reach required productivity.

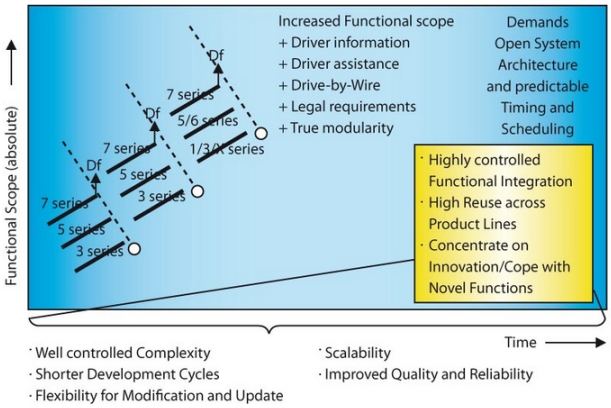

Figure 1 Main Requirements for System Architecture from an OEM's Perspective [Ru07].

AUTOSAR [He06] has established an industry standard for automotive E/E architectures. The standardized software architecture with well-defined module interfaces are key enablers for main AUTOSAR goals such as re-use, interoperability, portability, and especially the flexibility to distribute and (re-) map software components to different hardware topologies. However, the system performance changes significantly with the mapping decisions in terms of bus load, ECU utilization, buffering delays, and finally end-to-end signal timing. Hence, finding an optimized mapping –a key step in the envisioned AUTOSAR methodology– requires consideration of the timing and performance behavior of the involved software components and their potential mapping and implementation on the hardware. But defining a timing model for AUTOSAR is a challenging task [Ri06].

As a promising starting point, timing can be approached through the concept of "timing chains" that provide analyzable or predictable temporal relations between AUTOSAR-defined "observable events" [SR08]. These "observable events" represent actions such as "runnable starts" or "data sent" at which data or control is "handed over" from one component to another, e.g. between two software components, from a software component into basic software, from basic software to the bus hardware, etc.

As an example let there be a function called "Automatic Door Opener". This function opens the car's doors automatically, when the user's hand approaches the door knock and the car key is present in a close area around the car. The hardware key transmits a digital cryptographic key which is received and verified by an ECU. One typical high-level timing requirement for this overall function could be that the doors must be unlocked within 100 milliseconds after the hand has approached. This kind of high-level end-to-end-latency timing requirements can usually be found in specification sheets. This overall latency between the two *external* events "hand approached" and "door unlocked" can be divided into different segments using *internal* observable events, e.g. hand recognition, digital key verification and opening the lock. Components implementing the functionality of these segments can be provided by different suppliers and have to be integrated with respect to the high-level requirement mentioned above. It is the system integrator's responsibility to guarantee the fulfilment of the high-level requirements and thus himself formulate appropriate derived requirements for the suppliers.

In general, the idea of introducing hand-over points (HOPs) to structure timing models via timing contracts is not new [Br04, Ri06]. A general concept to add non-functional requirements (e.g. timing, safety) to component models is described using so called Rich Components in [Da05]. The authors propose to use a special constraint language to express timing requirements for input-output relations of component ports. Our approach follows a similar idea though it is applied to the AUTOSAR methodology and especially to the timing chain concept. In this concept, contracts can represent a specification of the observable event's timing properties at such HOPs (or groups of them). These could be formulated as guarantees, constraints, requirements, etc. In principle, the HOPs allow structuring and understanding what happens along a timing chain, whereas contracts allow abstractions from details of chain segments as a key means for OEMs and suppliers to communicate timing information without compromising IP protection. We will demonstrate these concepts using two prominent use cases in Sections 2 and 3.

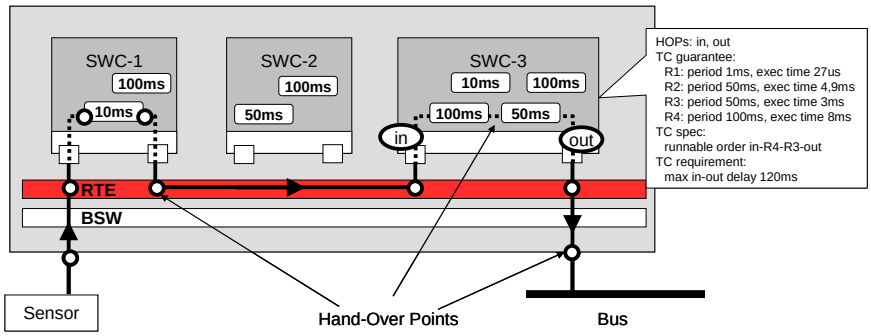

Figure 2 ECU with three SW components and several HOPs

Finally, HOPs and contracts must be defined such that the resulting model structure (and its parameters) supports analysis and verification because there is no point in modeling something that cannot be analyzed. Here, scheduling analysis concepts provide a good starting point because scheduling models are well structured and typically allow checking system-level properties based on component-level properties. Therefore, scheduling models and the idea of hand-over points and timing contracts fit together well. In Section 4, we use the SymTA/S scheduling analysis tool [He05] to generate gray-box timing chain segments from detailed models. Due to space limitations, we omit formal definitions but focus on the practical implications.

2 SW Supplier & ECU Integrator

In the first example, an ECU integrator deals with software components from several suppliers. To optimize the ECU configuration (step in AUTOSAR methodology) and verify the timing, the ECU integrator might use the following information about the supplied software, illustrated also in Figure 2:

- HOPs (internal): runnable start and end, possibly data send/receive and RTE calls
- specification: runnable periods and runnable order
- guarantees: runnable execution time, relative point in time where RTE is accessed; alternatively sequence of execution times and RTE calls
- requirements: max. jitter of RTE-calls, max input-output-delay of one runnable or the entire software component

From this data, the ECU integrator can map the runnables to tasks and configure the scheduling. Especially the contract requirements are useful to guide the decision making process and to optimize the runnable order, runnable-to-task mapping, etc. accordingly. Furthermore, the ECU integrator can analyze in detail the internal data-flow timing. The SymTA/S scheduling diagram in the bottom-left of Figure 3 illustrates the scope of information and analysis of the ECU integrator.

In a similar way, also basic software (BSW) suppliers can specify execution times for functions that the runnables call, possible depending on the context of the BSW call.

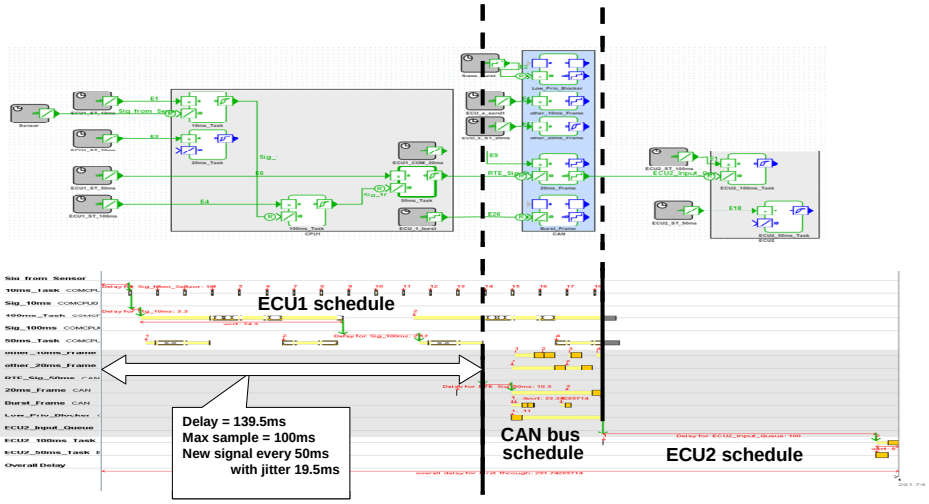

Figure 3 Detailed Schedule of End-to-End Timing

3 ECU Supplier & Network Integrator

Now, the ECU from Section 2 is supplied and integrated with another ECU2 to a CAN bus. It is the OEM's turn to build and optimize the network schedule. Here, the hand-over points (or internal observable events) are: signal and frame generation and reception time. For these, period and jitter are the most relevant attributes.

Analyzing the end-to-end delay along such a signal timing chain is a challenge [Ri07]. Figure 3 illustrates the SymTA/S implementation view (with tasks, frames, and signals) of a two-ECU system as well as the resulting SymTA/S scheduling diagram. We see that the first part of the end-to-end schedule equals the ECU-only timing as described in Sec. 2. In addition, the CAN bus schedule, the relevant RTE and COM signal delays, and the schedule of the receiving ECU is shown.

We want to note that the overall complexity of end-to-end analysis does not come from the priority-driven CAN arbitration. End-to-end timing analysis is also challenging for FlexRay. Despite its deterministic frame timing, the interaction of frame timing with signal timing and application task's scheduling, in combination with multiple options for synchronization, turns end-to-end analysis of FlexRay into a challenging problem. This shows that certain COM properties cannot be chosen with full flexibility but might be restricted by the interfaces to some extend. In CAN networks, this applies to the frame buffering strategy in COM/DRV.

4 Timing Chain Segments as Gray Boxes

In the above network example, the ECUs were integrated into the end-to-end analysis with all details about tasks, scheduling, etc.. In practice, however, it is very unlikely that ECU suppliers pass such detailed models to OEMs for several reasons incl. IP protection. On the other hand, not exchanging this information would significantly compromise the OEMs ability to perform any end-to-end analysis. As a solution, we envision a gray-box-like model for segments of the timing chain. These segments must hold all parameters needed to perform the end-to-end analysis, and can hide the rest.

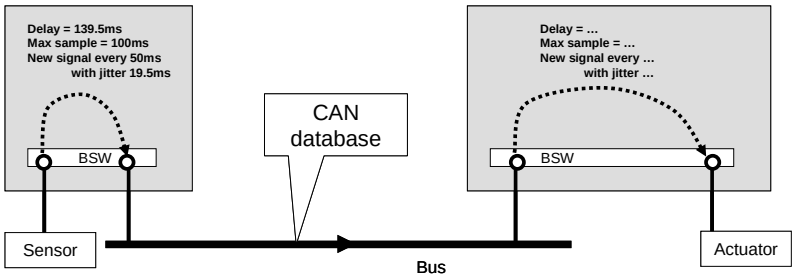

Figure 4 Network View with Gray-Box ECU Timing Models

The definition of HOPs is already a large step towards timing chain segmentation and gray-box models. Each and every HOP is a potential connector of a timing chain segment. In Figure 3, we have already outlined the key properties of the segment from the sensor through the ECU1 until the signal is produced:

- the period and jitter of signal generation times, derived from the 50ms task scheduling properties,
- the delay from the sensor to the signal output, derived from the end-to-end scheduling analysis of ECU1, and
- the maximum internal sampling period, i.e. 100ms, because this will affect the timing of new data at the output of the 50ms task

These are the most relevant properties to perform the end-to-end analysis for the rest of the system (here: the bus and the second ECU) at the same level of accuracy just as if all details were available. The ECU suppliers can provide such excerpts as gray-box timing segments to the OEM who can then do the end-to-end analysis of the entire timing chain from the sensor to the actuator. A corresponding gray-box system is shown in Figure 4.

An important question is at which level of accuracy such information shall be exchanged. This discussion is multi-dimensional. One dimension covers different phases of the design, where we could start with rough (inaccurate) estimates and refine them later. This lets us at least have basic sanity checks from early on. Another dimension is on the accuracy of the models itself. Mutually exclusive behavior, e.g. when we know that only one of two functions exhibits its worst case timing at the same time, is important. When ignored, analysis results are likely (too?) pessimistic [RO+07]. Furthermore, we must target analyzable models, i.e. models for which analysis strategies (and tools) exist. Other dimensions include modeling efficiency, comprehensibility, etc. Recent progress in bringing scheduling analysis theory to practice supports already a wide range of solutions [Ri07].

5 Summary and Conclusion

In order to fully roll-out the envisioned AUTOSAR methodology, the existing standard must be complemented by a reasonable view on timing and performance issues. Taking optimized mapping and scheduling decisions requires knowing the performance implications. Reasonable timing models can provide the necessary guidance for the two most important steps in the AUTOSAR methodology: system generation (incl. mapping SW components to ECUs) and ECU configuration (incl. runnable-to-task mapping and scheduling). Even more importantly, mapping and scheduling decisions can be based on

optimization strategies that also use timing-related quality metrics (e.g. minimization of the segment delay through a node or a COM-COM pair).

In addition to capturing system timing, effective distributed development of such systems requires a clear separation of responsibilities among OEMs, Tier-1 suppliers, and various application and basic software suppliers. In this paper, we have illustrated how timing interfaces and contracts along timing chains of observable events can improve this process significantly.

When chosen at the right places (hand-over points) in a chain, these contracts enable detailed timing analysis and verification locally and globally, which is important to understand and control the variety of timing effects within the implementation. Only then, designers can take reasonable actions to enforce more control on HOP timing and to fulfill constraints. Examples range from adjusting sensor sampling rates, selecting CPU speed ranges, or fine-tuning task or bus schedules. Furthermore, these HOPs are ideal candidates for abstracting from details and deriving gray-box timing chain models to be exchanged between the involved parties without compromising IP protection.

In this paper, we have shown some very relevant use cases demonstrating the importance and application of timing interfaces. From a timing viewpoint, systematic HOPs and gray-box models provide essential means for the mentioned use cases. Furthermore, we have shown that –due to their structure– scheduling analysis models provide the necessary abstractions, enable gray-boxing, and support system-level timing analysis.

Our idea of a timing interface introduced in this work fits well as an addition of AUTOSAR. But still AUTOSAR needs to be augmented with a basic timing model to enable the application of such timing interfaces. An initial timing model is currently developed within AUTOSAR for the next release. Symtavision is further contributing its knowledge to the EU-funded TIMMO project [Je07].

References

[Br04] J.-Y. Brunel, M. Di Natale, A. Ferrari, P. Giusto, L. Lavagno. SoftContract: an Assertion-Based Software Development Process that Enables Design-by-Contract. In Proc. Design, Automation, and Test in Europe (DATE), Paris, France, 2004

[Da05] W. Damm, A. Votintseva, A. Metzner, B. Josko, T. Peikenkamp, E. Böde, Boosting Re-use of Embedded Automotive Applications Through Rich Components. In Proc. Foundations of Interface Technologies, Foundations of Interface Technologies, San Francisco, USA, 2005.

[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

[Ri06] K. Richter. The AUTOSAR Timing Model - Status and Challanges. ARTIST2 Workshop Innsbruck. March 2006

[Ri07] K. Richter. How OEMs can get Suppliers On Board for Designing Extensible Networks. In Proc. Embedded World Conference, Nürnberg, Germany, February 2007.

[Ru07] M. Rudorfer, T. Ochs, P. Hoser, M. Thiede, M. Mössmer, O. Scheickl and H. Heinecke. Realtime System Design Utilizing AUTOSAR Methodology. Elektronik automotive, Special issue on “AUTOSAR”, October 2007

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