Session 1 – AutomotiveApplications

Chair: Christos Koulamas

Rapporteur: Jean-Dominique Decotignie

Session Summary: Automotive Applications

Rapporteur:
Jean-Dominique Decotignie
Swiss Center For Electronics and Microtechnology (CSEM)
Neuchâtel, Switzerland
Jean-dominique.decotignie@csem.ch

1. Introduction

The session had only one paper but on a very interesting topic, the gap between researchers and practionnners in the domain of real-time networking in particular in the automotive domain. This is a very important issue as networks today become the integration point between different automation components of the car.

2. The presentation

The paper is based on practical experience with engineers developing automotive applications. It explains that, despite nearly 2 decades of research papers on the automotive networks, little of the "theoretical" results is used in practice. Reasons may be found in the difference between the "clean" models used in the papers and the actual hardware employed. The same apply to software stacks deployed on the hardware that may not implement any idea of priority thus rendering void the nice properties of networks such as CAN.

The automotive industry tries to enhance the development process by defining common architectures such as AUTOSAR. The paper shows that temporal aspects are not the main concern of such initiative. Furthermore, AUTOSAR leaves too much freedom and it seems very difficult to find a temporal model that can be applied to the system. For instance, different interaction models, client server, periodic, …, may coexist and it becomes difficult in such a context to provide a clear definition of deadlines.

3. Discussion

For the presentation and the discussions, it is clear that bridging the gap between researchers and practionners is desirable but far from easy. It is also obvious that this should be the task of the research community although there is a need to change the minds in the companies. In particular, studies should address the following aspects:

- the analysis should include all the software aspects;
- this must include a model of the temporal behaviour of the ECUs (Electronic Control Units) that interact through the communication network; this may go as far as looking at the behaviour of the operating system or kernel used in the ECU;
- using the "theoretical results" should be as easy as possible and this may be related to finding the right level of abstraction;
- the model should allow to define simple things such as deadline in a common manner;
- implementing (in software) the "theoretical" models should be easy.

4. Conclusion

This presentation opened a number of possible research venues and let us hope that the subject will be at the agenda of future RTN workshops.

Applying Real-Time Network Research in the Automotive Industry: Lessons Learned and Perspectives

Dr. Kai Richter[1], Dr. Marek Jersak[1], Prof. Dr. Rolf Ernst[2]

1: Symtavision GmbH
Frankfurter Straße 3b
38122 Braunschweig, Germany
{richter|jersak}@symtavision.com

2: Institute of Computer and Communication Network Engineering
Technical University of Braunschweig
Hans-Sommer-Straße 66
38106 Braunschweig, Germany
r.ernst@tu-bs.de

Abstract

This paper addresses the still very large gap between the research community and the industry with respect to the application of real-time networks analysis. As a university spin-off providing scheduling analysis solutions, we have made several controversial experiences with the technology transfer that we would like to discuss during the workshop. Key examples from practice and a look into the industrial process of designing –and the way of thinking– shall help structuring the discussions.

1. Introduction

The area of real-time systems research including networks is a very active field for more that 30 years. Countless publications are available such as on analyzing and optimizing the timing behavior of CAN (controller-area network) communication, a widely used standard in the automotive industry. The number of contributions concerning FlexRay, often promoted as a CAN "successor", is growing, too. With the integration of more and more networked functionality in cars, network timing and performance has become a critical bottleneck in automotive architecture design, with a direct impact on design time and cost. As optimal network design and configuration requires reliable analyses and good optimizations, one could conclude that car manufacturers should be eagerly adopting technical contributions in the field of real-time networks research. However, the willingness to do so is surprisingly low. But why is that?

The reasons are multifaceted. Over the past years, we have been continuously facing that question in a number of projects in the automotive industry [1], from car manufacturers to tier-1 and software suppliers to service providers. As a university spin-off that now develops and markets the SymTA/S scheduling analysis and optimization tool suite and services, there have been interesting technical "surprises" that might appear as a key reason. In fact, there is very often a mismatch between well-defined theoretical models and the industrial practice. Additionally, practicability concerns and political, cultural, and economical reasons add to the dilemma, as they complicate convergence of both parties. This paper outlines key experiences that we have made with respect to the issues mentioned, some of them have been presented earlier [2].

2. Model Mismatch

Researchers and designers have significantly different focus. Designers have to produce something that works within a reasonable time frame. Hence, they stick to established approaches, even if it requires an enormous effort to finish the task at hand more or less on time. Quite to the contrary, researchers use their freedom to consider a variety of conceptual options to develop a consistent and well-structured theory, and then write it down. Eventually, researchers and designers are worried about the same general topic, for instance, network integration, and start talking to each other. This often reveals a different view on "the problem"; different with respect to importance, model soundness, and analyzability. We will briefly outline two illustrative examples of such "surprises".

a. CAN Example

The first example is the use of queuing strategies in CAN networks. The medium access in CAN is based on a priority-scheme. CAN frames that compete for the bus are scheduled according to their priority, coded in the CAN Id. Waiting frames on an ECU (electronic control unit) are buffered to be sent later. To no surprise, the big majority of formal methods to analyzing such systems assume that the buffering strictly follows the priority-driven strategy of CAN, as this is (!) consistent with the protocol itself. However, CAN implementations contain several software and hardware buffers. Each uses its own access strategy, including FIFO instead of priority-ordered queuing. FIFOs undermine the CAN protocols inherent access strategy and thus the available analysis techniques. Figure 1 allows comparison between two such schedules. Priority-queuing on the left leads to schedules that can be analyzed with a static-priority analysis technique, while FIFO queuing significantly complicates timing behavior and reduces analyzability.

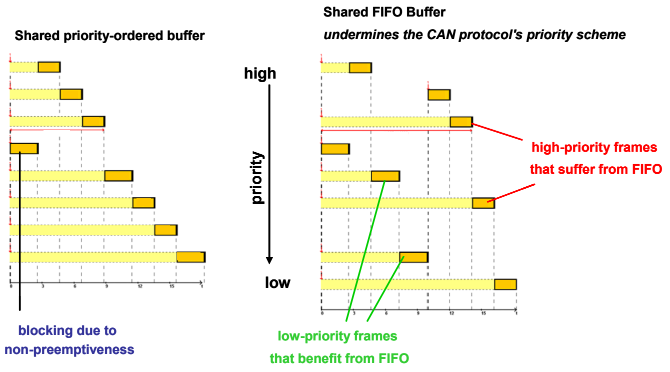

Figure 1 Effects of Priority and FIFO queuing

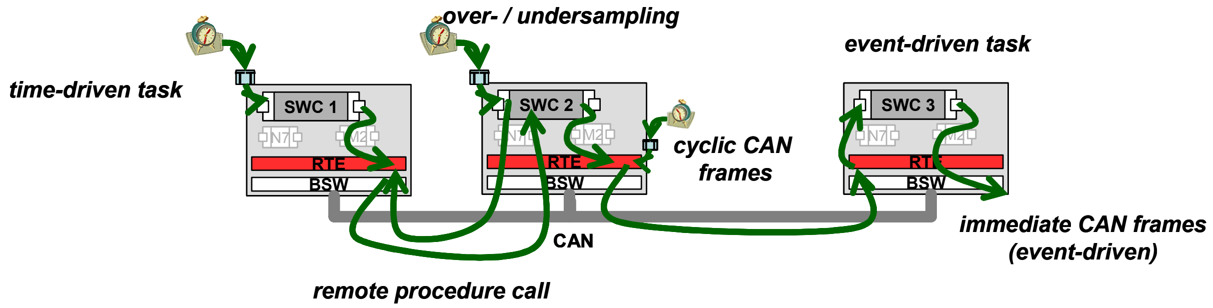

Figure 2 Causality Chains in Automotive Implementations

Other real-world mechanisms further complicate analysis. In particular, so called “overload management mechanisms” skip frames that wait in the buffers “too long”, thus violating another assumption (fixed load) of typical scheduling analyses.

It is amazing to observe what designers do for lack of reliable analysis. Often significantly more messages than actually required are sent. The assumption is that allowing "N out of M" messages to get lost is a way to “guarantee” that a minimum number of messages get through. Obviously this increases bus load in the typical case and is thus counter-productive to the desire to reduce bus load to make room for more messages required for novel vehicle functions.

How should we approach this discrepancy between analysis and the real-world? The researcher might say: “This is not analyzable! Go redesign your system and come back to me!” The designer might say: “If you do not develop an analysis for exactly my problem, your kind of research is useless!” It is clear that neither party has any benefit of insisting on his/her position.

b. AUTOSAR Example

The second example illustrates another important type of model mismatch at a higher level of communication. With the increasing distribution of functions over several ECUs in a car, the importance of end-to-end timing (and deadlines) is also increasing. Industrial standardization efforts such as AUTOSAR have already defined models for capturing such “timing chains” composed of communicating “software components”, illustrated in Figure 3.

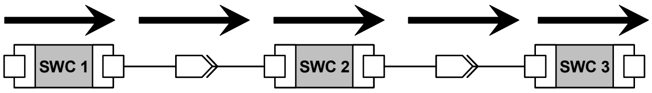

Figure 3 AUTOSAR View on "Timing Chains"

Similar models are known from data-flow theory, where clear semantics relate the execution of nodes (here: software components) with timing behavior of the stream. “Surprisingly” though AUTOSAR has not yet defined such relations. Quite to the contrary, the actual timing of software components is mostly left open. Additionally, there exist several valid communication semantics including client-server (remote procedure call), periodic sampling including under- and over-sampling, polling, and event-driven. This leads to a variety of possible “causality chains” in the actual implementation that can be subject to analysis. Figure 2 shows examples for these causality chains through the layered software defined by AUTOSAR.

What does “end-to-end timing” mean in absence of semantic definitions? Again, the lack of a “common ground” leads to a mismatch between the work of researchers and the challenges system designers face, and both proceed in isolation.

3. AUTOSAR Background

It is important to understand that the primary goal of AUTOSAR is not to solve timing problems in particular. AUTOSAR rather defines a software infrastructure for application and basic software, illustrated in Figure 4. The goal is to be able to exchange parts of the system’s software without rebuilding everything. This shall enable modularity, scalability, transferability and re-usability of software among projects, variants, suppliers, customers, etc.. Hence, timing is not in the center of AUTOSAR but has later been recognized as an “important issue” that requires further consideration.

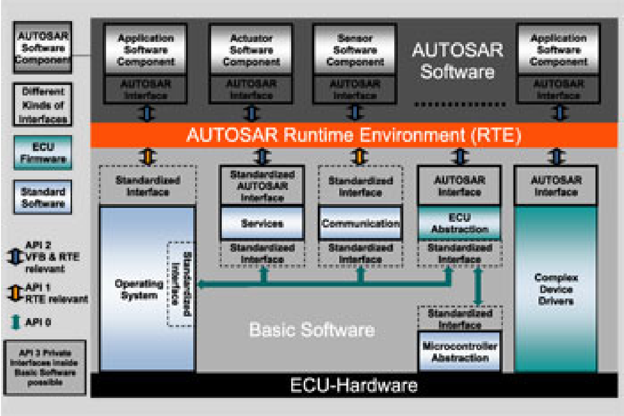
AUTOSAR Software Component
Different Kinds of Interfaces
ECU Firmware
Standard Software
API 2 VFB & RTE relevant
API 1 RTE relevant
API 0
API 3 Private Interfaces inside Basic Software possible
Application Software Component
Actuator Software Component
Sensor Software Component
AUTOSAR Software
Application Software Component
AUTOSAR Interface
AUTOSAR Runtime Environment (RTE)
Standardized Interface
Standardized AUTOSAR Interface
Services
Communication
ECU Abstraction
Operating System
Complex Device Drivers
Basic Software
Microcontroller Abstraction
ECU-Hardware

Figure 4 Standardized AUTOSAR Software

Although the AUTOSAR specification is not open to the public, the website www.autosar.org provides a quick overview and few papers [3]. However, AUTOSAR borrows many key concepts from the OSEK/VDX standard that is available through www.osek-vdx.org. These documents define a layered software-architecture with many APIs in many configurations but only few semantics. With respect to example b), especially the communication layer with its various configuration options such as “triggered vs. pending signals” that are sent through “periodic, direct, or mixed frames” [4] is a major source of network and system-level timing complexity. The standard further lets open the implementation of the lower-level drivers. We have seen in example a) that queuing strategies in particular can make the difference. Finally, also the OSEK OS [5] standard knows of several task types and activation mechanisms, complicating the analysis of schedules and timing chains (example b) further.

4. Practical Issues

In addition to a common technical ground, designers also need to be able to embed a researched technology into their everyday design-flow. Based on the feedback we have been receiving from a variety of designers, this in particular requires:

1. generating or obtaining the data needed for analysis (be it by definition, measurement, test, or simply asking the right people)
2. having a specific strategy when and how to apply the technology
3. interpreting the results and consequently taking decisions.

All this in a reasonable amount of time, after a reasonable amount of training on that technology.

If the technology appears too complex, designers will ignore it. If input data is not readily available, they can't use it, and if using the technology takes longer than finding a sub-optimal manual solution, it will be considered useless, again.

We highlight this, as researchers (rightfully) tend to do work that is "elegant" or "systematic" in itself without paying too much attention to practical issues.

5. Supply-Chain Issues

Specifically car manufacturers nowadays have to cope with an increasing number of network real-time problems that are fully new to them. Historically, car manufacturers designed mechanical parts. The electronics parts including the software were, for a long time and still, supplied externally. Hence, the OEMs became used to their suppliers solving the technical problems related to software.

Now, the OEMs still do not develop large parts of the software. However, as a result of function distribution, the network turns into the center of many integration efforts for which the OEM is responsible. As illustrated in the CAN example, the network timing depends not only on the protocol but also on driver hardware and software. And even though the OEM controls many network parameters such as topology, speed, and frame priorities, the drivers are often not in the OEM's area of responsibility.

6. Possible Solutions

The supply-chain communication between OEMs and suppliers will have to evolve. As ECU implementation possibly affects network timing, relevant data may have to be disclosed by the suppliers. From the other perspective, OEMs could impose ECU timing requirement on their suppliers that they know will satisfy assumptions on the timing of the communication infrastructure.

This leads to the idea of establishing timing contracts between OEMs and suppliers for each "module" or "component" that is designed individually but contributes to the overall system timing.

Finding a right strategy is difficult. In order to be accepted

- Responsibilities and scope must be clearly defined, and must (more or less) 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 such as AUTOSAR, this will have a dominant impact on the structure of the analytical model.
- A suitable timing analysis methodology must be in place. Based on the structure just mentioned, the analytical possibilities will to a large part define the parameters of the model, 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 at a particular design stage, and the required effort.
- Any analysis methodology must allow engineers to reason about their decisions systematically. 100% accuracy may not be needed if only the results are significantly better than "gut feeling".

An important step is to further standardize and "homogenize", in order to reduce complexity. Today some OEMs have defined a "standard core" with predefined OS and driver-level concepts which every supplier must implement. This ensures more predictable timing of the communication infrastructure; and better configurability. AUTOSAR has helped to define standard interfaces between components at various granularities and levels of abstraction. First reference implementations show that the interfaces work.

However, the current view is still very function- and software-centric; AUTOSAR version 1.0 does not include timing, and hence does not tackle timing-related integration issues. Furthermore, the standard does not contain clear guidelines how to use the standardized technology. Therefore, it is not clear how to establish a ready-to-use analysis approach. Guidelines along the "standard core" approach may therefore be needed in order to cover a significant number of problems with a suitable timing analysis methodology.

In several projects with OEMs, tier-1 and -2 suppliers, we have seen that each particular partner is in fact capable and willing to apply a certain amount of timing analysis, if only the scope is suitable, the analysis can be performed efficiently, and they see a real value for them.

For instance, OS and basic software suppliers can determine the latency of service routines, driver functions, and disclose key mechanisms such as queuing strategies. Function designers can use measurements or formal analysis to obtain execution times of their functions, along with amount of communicated data. Similarly, ECU suppliers can do more precise and systematic measurements to generate the data required for thorough scheduling analysis, and they can build timing interfaces to the bus, specifically with respect to dynamic driver interrupts. This is already a large step towards supplier-OEM timing contracts. Finally, OEMs have started to use their knowledge about the "standard core" to gather information about key queuing mechanisms used in their systems. From the knowledge about these mechanisms, together with the software supplier's data and the dynamic ECU-network timing interface, we have established and solved scheduling models that particularly support OEMs in comparing the performance and robustness of several configurations, without requiring any of them to understand the full picture.

7. SymTA/S Review

Our own technology, SymTA/S, has been originally developed at the Institute of Computer and Communication Network Engineering [6] and is based on the idea of compositional scheduling analysis. In contrast to the holistic approaches, SymTA/S allows direct re-use of the host of existing single-processor scheduling techniques such as RMA/DMA, EFD, TDMA, RR, etc.. Details are not individually cited here but can be found in a SymTA/S overview paper [7].

SymTA/S captures system-level dependencies through event models at the interfaces between locally analyzable components. This gives structure to the model and protects IP internal to the components, be it software components (tasks), ECUs (CPU resources) or buses. Furthermore, we have developed configurable analysis libraries tailored to the concepts defined by OSEK, AUTOSAR, CAN, and we are currently working on a compliant library for FlexRay.

By keeping the first-class citizens of the analytical model small and in line with the established industry system view, the involved parties can in fact establish timing contracts that they can oversee. And the compositional approach enables establishing a system-level analysis from such black boxes.

Furthermore, using a "tool box" of technologies from real-time systems research rather than a single approach, allows quick extension and customization of the analysis, a prerequisite for meeting key requirements mentioned in Section 6. We have successfully applied SymTA/S in several industry projects with customers [1, 8], and are constantly extending it with academic partners.

The "pure" analysis is supplemented by a set of productivity plug-ins. An exploration module [10] uses genetic algorithms to find optimized system configurations. Sensitivity analysis [11] is used to detect and avoid critical hot spots in the design. Finally, the technology has also been used in a multi-supplier risk management system [12].

8. Summary

The gap between research and industry is still large in the area of real-time networks. Two key examples have shown that technical barriers are only one reason. Practicability issues, supply-chain communications and other strategic or even political decisions are other reasons. In this paper, we have outlined a set of requirements and possible solutions. We have further seen that we could already apply some of them successfully in practice using our SymTA/S tool suite. Key to this is that all involved parties must approach each other within a bounded scope of technical problems and clear goals. We ultimately believe that, after some time, designers will themselves distinguish a technically sound (and elegant) solution from a less systematic one. They will do it to reasons of analyzability and safety rather than elegance. However, any such successful cooperation between industry and research, at best with evident benefits, helps fostering the appreciation of real-time networks research.

This step-wise approach still requires a suitable methodology, along with models, which have to be developed. This also includes re-thinking the roles of OEMs and suppliers and their communication along the supply chain, possibly leading to an engineering evolution for individual partners, and a cultural change in a new multi-supplier design process management.

References

[1] Kai Richter, Rolf Ernst. Real-Time Analysis as a Quality Feature: Automotive Use-Cases and Applications. In *Embedded World Conference*, Nürnberg, Germany, February 2006.

[2] Kai Richter, The AUTOSAR Timing Model—Status and Challenges, *ARTIST2 Workshop "Beyond AUTOSAR"*, Innsbruck, 2006

[3] AUTOSAR Partnership. AUTOSAR – Current results and preparations for exploitation. *7th EUROFORUM conference Software in the vehicle*. 3-4 May 2006, Stuttgart, Germany

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

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

[6] SymTA/S Project. Institute of Computer and Communication Network Engineering, Technical University of Braunschweig, Germany, www.symta.org

[7] Rafik Henia, Arne Hamann, Marek Jersak, Razvan Racu, Kai Richter, Rolf Ernst. System Level Performance Analysis - the SymTA/S Approach. *IEE Proceedings Computers and Digital Techniques*, 2005.

[8] Kai Richter, Marek Jersak, Rolf Ernst. How OEMs and Suppliers can tackle the Network Integration Challenges. In *Proc. Embedded Real-Time Software Congress (ERTS)*, Toulouse, France, January 2006.

[9] Arne Hamann, Marek Jersak, Kai Richter, Rolf Ernst. A framework for modular analysis and exploration of heterogeneous embedded systems. *Real-Time Systems*, volume 33, pages 101-137, July 2006.

[10] R. Racu, M. Jersak, and R. Ernst. Applying sensitivity analysis in real-time distributed systems. *In 11th IEEE Real-Time Technology and Applications Symposium (RTAS)*, San Francisco, USA, 2005.

[11] J. Kruse, T. Volling, C. Thomsen, R. Ernst, and T. Spengler. Towards Flexible Systems Engineering by Using Flexible Quantity Contracts. In *Proc. Automation, Assistance and Embedded Real Time Platforms for Transportation (AAET*), Braunschweig, Germany, 2005.