

A Virtual Platform for Architecture Integration and Optimization in Automotive Communication Networks

Razvan Racu, Rolf Ernst

Institute of Computer and Communication Network Engineering
Technical University of Braunschweig
Hans-Sommer-Straße 66
D-38106 Braunschweig, Germany

Kai Richter, Marek Jersak

Symtavisio GmbH
Frankfurter Straße 3b
D-38122 Braunschweig, Germany

Copyright © 2007 SAE International

ABSTRACT

Systems and network integration is a major challenge, and systematic analysis of the complex dynamic timing effects becomes key to building reliable systems. Very often, however, systematic analysis techniques are (considered) too restrictive with respect to established design practice. In this paper we present lessons learned from real-world case studies, in which OEMs have used the new SymTA/S scheduling analysis technology to evaluate different network choices with minimum effort. Thanks to its flexibility and supplier independence, SymTA/S can be applied in non-ideal situations, where other, more restricted technologies are inherently limited. Finally, we put the technology into relation with ongoing standardization activities.

INTRODUCTION

The increasing application complexity, together with a strong time-to-market pressure, requires a massively parallel design of systems, whether in automotive, avionics, multimedia, or telecommunications industries. The supply-chain often contains hundreds of companies that design their individual "components" based on requirement definitions from the OEMs (Original Equipment Manufacturer) or Tier-1 suppliers.

Systems integration is a major challenge. Dynamic component interactions result in a variety of non-functional performance dependencies due to scheduling, arbitration, blocking, buffering etc. These can lead to hard-to-find timing problems, including transient overload, buffer under- and over-flows, and missed deadlines. There is a growing need for methods that allow to safely eliminate such problems up-front.

In an ideal scenario in which the system is designed from scratch with full flexibility, synthesis could be used to systematically translate well-specified, formalized sets of requirements into homogeneously optimized network configurations, thereby maximally exploiting the design freedom. Even though this ideal approach appears promising, it suffers from two critical practical restrictions.

First, each synthesis is typically constrained to a specific communication architecture and a subset of communication mechanisms. In practice, the design is far more complicated. Many of the mentioned timing problems directly result from the integration of re-used components and the resulting design style, in which platforms evolve heterogeneously over several product variants and generations. This heterogeneity of systems and design processes (with change-request iterations) challenges the applicability of synthesis, and fully changing the design process is no practical option for OEMs.

Secondly, very often the specifications are found to be incomplete, which counters the synthesis need for a complete specification. Standardization could help increasing the awareness of timing data and finding a common language. Many ideas have been proposed towards a more requirements-based or contract-based design, also in the area of automotive software [1].

However, standardization is slow. Even the AUTOSAR standard [1] still lacks aspects of timing and performance, although these have been recognized as a major challenge for system integration. Finding a common ground seems difficult because of the heterogeneous roles in that multi-supplier industry, and because most players in industry have only little

experience with timing and performance analysis. These are among the major reasons why synthesis and requirement-based design have not yet been widely applied in industry.

We have extensively researched the role of performance and timing analysis in system-level design, in the SymTA/S project [13], and we have adopted and extended a host of theoretical contributions to meet industrial requirements. Key to the SymTA/S approach is its flexibility to adapt to heterogeneous, real-world design environments, and its inherent ability to incorporate legacy components, ensuring applicability far beyond those of the ideal scenarios depicted above.

We have successfully demonstrated the SymTA/S technology in a number of industrial projects. In this paper, we will summarize the application of SymTA/S (Symbolic Timing Analysis for Systems) in the area of automotive network dimensioning, the very center of all integration efforts. We demonstrate how network reliability and the overall integration process can be significantly enhanced with minimum effort, thereby respecting the established business models along the industrial supply-chain with its non-ideal heterogeneous design scenarios.

We will start with identifying the challenges that OEMs and suppliers face, and provide a quick overview about the foundations of related analysis fields.

THE NETWORK INTEGRATION PROBLEM

Network integration is carried out by an OEM who determines bus topology, speed, number of nodes and frames, and finally the configuration, e.g. the assignment of priorities or time slots to bus frames. The decision making process typically includes the following questions: Is the network (temporarily) overloaded? Which frames can get lost, and how often? Can more ECUs (electronic control units) be connected, and how many, without overloading the bus? How about diagnosis and ECU flashing? How to integrate legacy ECUs and software most effectively?

Answering these questions requires understanding the intricate effects that individual decisions might have on the overall performance and timing. Furthermore, it requires a systematic procedure including appropriate supply-chain communications in terms of data sheets and requirements specifications.

Interestingly, things look quite different today. Simulation, prototyping and test are established common practice. However, they cannot systematically provide corner case coverage and are thus not suitable to reliably detect all bottlenecks. Therefore, architects very often favor less efficient, conservative designs. For instance, conservatively allowing N out of M frames to get lost is not an unusual way to “guarantee” that a minimum number of frames get through. A rather paradoxical approach, since sending significantly more

frames than actually required further increases bus load and thus further increases the number of lost frames. Detecting and reducing such inefficiencies, requires knowing how frame loss can be reliably analyzed and bounded. The lack of systematic procedures currently prevents OEMs from thorough optimizations, and overly conservative approaches are common practice.

QUICK REVIEW

LOAD ANALYSIS IS NOT ENOUGH

Although very simplistic, the bus load model is still among the most popular analytical models for bus analysis in practice. For each frame, multiply the frequency of a frame ($1/\text{period}$) with its length (including the protocol overhead), build the sum over all frames, and finally divide it by the network bandwidth. The result is the average network load, often called utilization, given in percent of the available bandwidth.

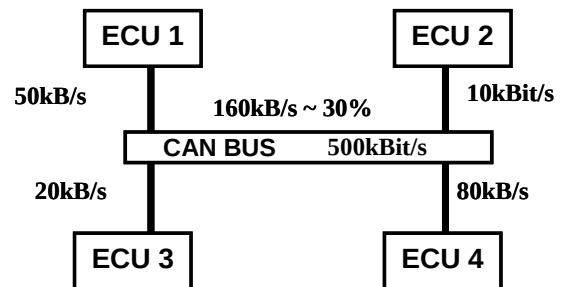


Figure 1 Simple load analysis example

Figure 1 shows an example. Four ECUs produce bus traffic that accumulates on the shared bus. The resulting total traffic is 160kBit/s which represents a load of approximately 30% on a 500kBit/s CAN bus.

Such load models are popular but there is, interestingly, much variation among the OEMs in defining a critical bus load limit; we have seen numbers as low as 30%, up to more than 60%. Why is that? Clearly, increasing the load means better resource utilization which translates into promising cost savings. The load model alone, however, does not say under which conditions deadlines can be met or if buffers may overflow, and should, therefore, be carefully used.

The correctness of a bus configuration depends on more parameters, especially with respect to the subtle dynamic effects that average load models can not capture. In fact, network reliability depends as well on the integrity of each supplied ECU as on the interactions among all ECUs. In particular, dynamic component interactions can result in performance dependencies and hard-to-find timing problems, including transient overload, jitter, burst, and missed deadlines. Figure 2 shows such a complex frame sequence with frame burst

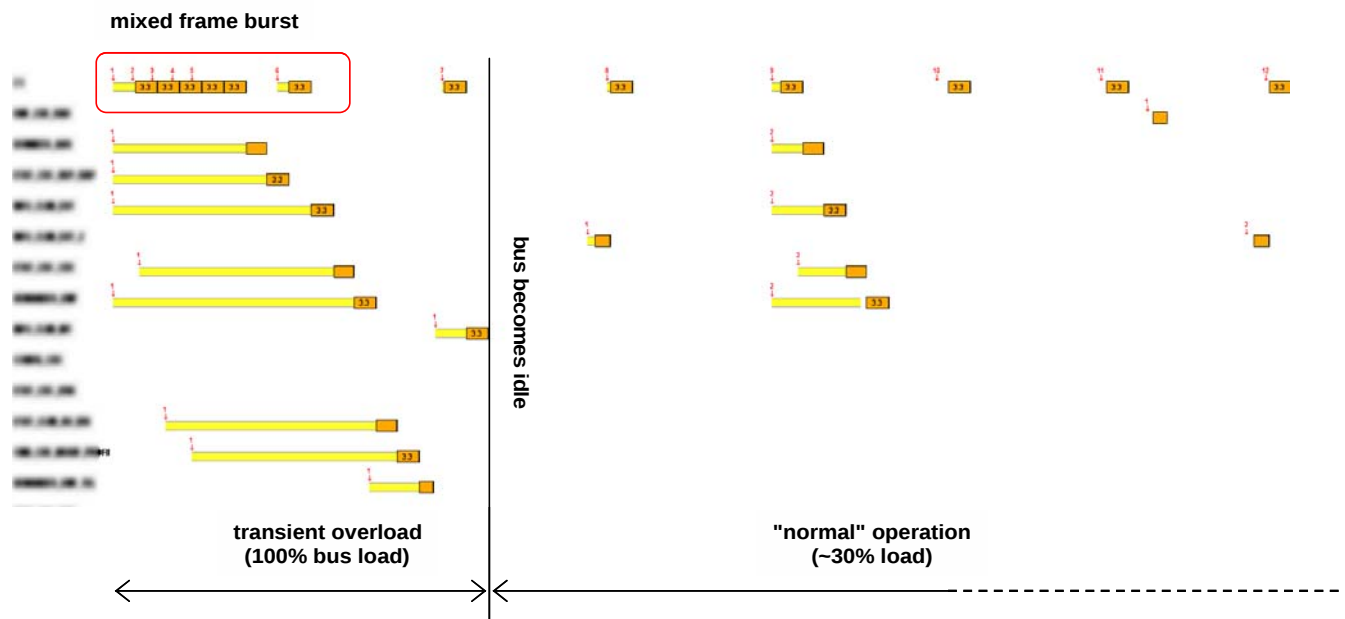


Figure 2 Frame bursts result in a complex communication pattern

and dynamic sender patterns during transient overload. More than once this has led to late stage integration errors and significant revenue loss in the automotive area.

SCHEDULABILITY ANALYSIS

Finding bottlenecks requires the complex dynamic communication patterns to be analyzed. There are sophisticated and proven methods available from real-time scheduling theory to determine and analyze such patterns automatically, from which we can only cite few landmark contributions [8, 12, 14, 6, 5]. In this paper, we will not introduce the theoretical foundations of this work. We rather emphasize the practical impact of having such techniques.

Synthesis-Based Approaches

Some methods such as Rate- and Deadline-Monotonic scheduling exploit certain timing properties and propose automatic priority assignments, which have been proven optimal under specific preconditions [8]. Mentor's Volcano Network Architect [16] is one such synthesis framework that automatically creates optimized communication matrices for CAN, based on Deadline-Monotonic Scheduling [8]. Such synthesis is well applicable when designing from scratch new, homogeneous systems with a predefined software architecture because, then, the synthesis algorithms can maximally exploit the available design space –an ideal precondition.

This applicability, however, decreases in more constrained design environments or when certain architecture decisions have already been taken, e.g. dedicated third party basic software (e.g. drivers), certain communication types, or specific buffering mechanisms. Furthermore, synthesis needs guidance, but it is not always clear what the actual requirements of

frame and signal timing are. Often, this lack of information heavily reduces synthesis effectiveness, along with its applicability.

Furthermore, frame and signal timing specifications need to be known. In effect, the current synthesis based approaches build an automotive software system around an analyzable and synthesizable proprietary software architecture and design method. That raises the issues of portability and future extensibility.

Analysis-Based Approaches

Other methods focus on the analysis aspect, also for non-optimal systems. The goal is to directly provide detailed data such as response times for any given system specification. For instance, to guarantee that a frame X will never get lost (overwritten in the sender's buffer), its maximum response time must not exceed its minimum inter-arrival time (the deadline). In automotive network design, frame deadlines of 10% to 20% of the frame period are commonly considered safe.

Calculating the response times requires consideration of the protocol-specific behavior of the CAN bus. Higher-priority frames can significantly delay lower-priority ones, frame jitters further distort the timing behavior, the controller type (basicCAN, fullCAN, etc.) influences the order in which frames are sent, and bus errors can lead to retransmissions and additional load.

A key property of such analysis techniques is that they find and evaluate the critical situations automatically without user interaction, provided that a minimum amount of bus configuration, i.e. the frame IDs (priorities) and frame length, is known. Further information about dynamic send patterns, the interface queues, and an error model further improve the accuracy and expressiveness of the results. Figure 3 structures this information into bus-related, ECU-related,

and models for error and flashing/diagnosis. Once this data is available, one can use analysis to provide system timing properties without the need for requirements.

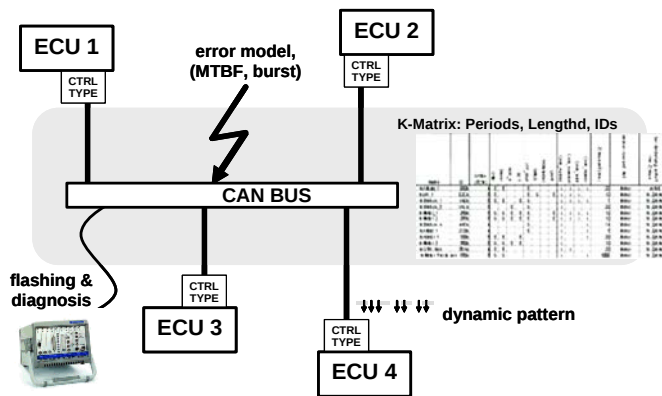


Figure 3 Information supported by reliable schedulability analysis

The SymTA/S Approach

With SymTA/S, the designer follows an analysis based approach. SymTA/S is based on a modular mathematical model which scales to include software architectures from different sources and suppliers. It capitalizes on the host of work in scheduling theory. For any new architecture, a new model can be easily developed or adapted, such that the tool can follow the development of automotive architectures over time. As an important example, SymTA/S is also capable of considering frame offsets and task-dependencies that will play a dominant role in the experiments later in this paper.

Then, we embedded the powerful analyses into a flexible design framework that poses very few restrictions on the systems and design processes. This framework supports incomplete requirements, missing data, and heterogeneous system configurations. Thanks to this flexibility, SymTA/S can be applied in non-ideal situations that were not amenable to effective and safe analysis, so far.

As an example, a critical and highly dynamic communication sequence identified by SymTA/S is presented in Figure 2. It shows regular periodic frames and a frame burst that results from dynamic sender patterns and/or bus errors. It can further be seen that, due to frame offsets, not all frames require the bus simultaneously.

In practice, however, very often only part of data is available to the OEM, usually in the form of a so called communication matrix, covering the period, length, and priority (CANId). The gray area of Figure 3 illustrates "scope" of OEM information.

The important dynamic influences are often not available in detail. These include so called *mixed* and *direct* frames that can dynamically appear at virtually any time within the periodic frames (names according to AUTOSAR and OSEK-VDX definitions of *frame transmission mode*). Similarly, the queuing strategies influence frame ordering in the COM layers and drivers, and can "undermine" the priority-driven nature of the CAN protocol. This data is part of the ECU implementation and typically not disclosed to the OEM.

The same holds for frame offset values. Such offsets define local phase relations between frames sent by the same ECU. The rationale behind using offsets is to "balance" the production of frames. Roughly speaking, offsets produce gaps (idle times) in the schedule that other frames can exploit. This balances the ECU interruptions by COM, and it also balances the bus load. Hence, offsets have a positive effect on bus load and, subsequently, response times. But again, such local offsets are typically defined by the ECU supplier and not disclosed to the OEM, nor asked for by the OEM.

From our experience, it seems that such non-ideal situations with lots of "unknowns" are the typical ones, while precise timing requirements and design from scratch are exceptions.

So, can analysis technology really help when data availability is a major concern? In fact, it can, if it is only flexible enough, as we will see by the end of the next section.

CASE STUDY

We have applied the SymTA/S technology in a number of studies for automotive OEMs, where several ECUs are connected to a bus, sending and receiving a total number of 100 frames and more. In all studies, we imported the length, CAN id (priority), and the period of each frame from a (customer-specific) *communication matrix* or *dictionary*. We knew frame offsets of only few ECUs, typically the gateways that are under control of the OEM. For other ECUs, we had no offset information. Information about dynamic patterns of *mixed* and *direct* frames was mostly missing.

Due to the lack of this important information, we typically conducted a set of experiments with SymTA/S, each based on different assumptions on the missing information. Simulators or prototypes were not required, as SymTA/S is based on mathematical techniques known from real-time scheduling theory. However, the SymTA/S analysis libraries are tailored to the specific real-world mechanisms of protocols (and OSEs), and thus, the results have a quite high degree of accuracy.

In the case study presented here, we first ignored all offsets. We considered all frames as being asynchronous and determined their response times. Obviously, such simplifications (no offsets) lead to overestimations with limited practical relevance.

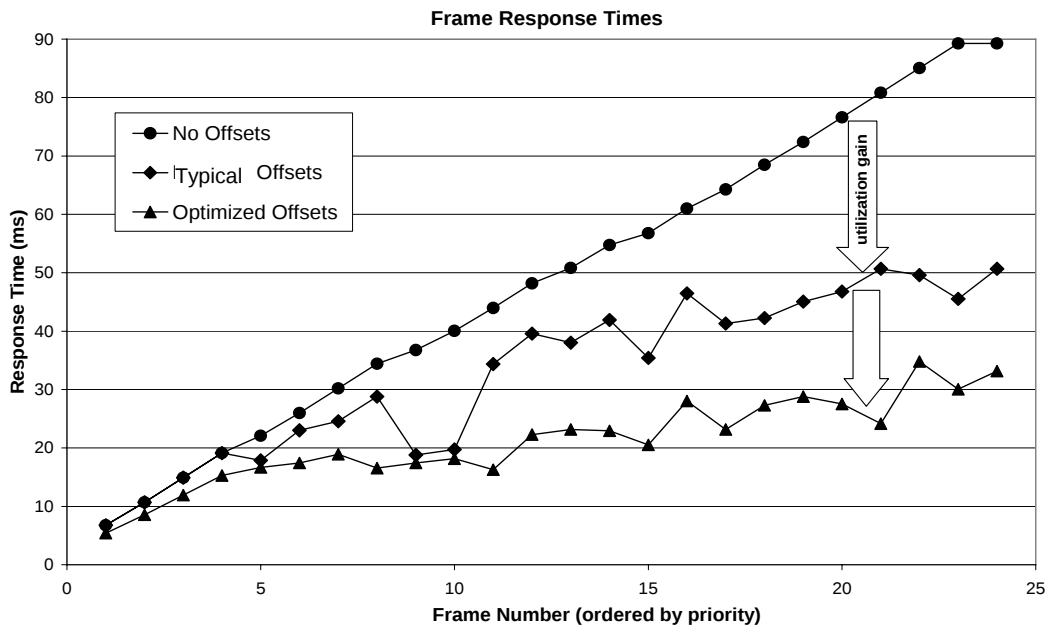


Figure 4 Reducing frame response times through offset optimization

However, the fact that we were able to carry out such "what-if" observations within minutes, without any simulation, prototype or test equipment is very important.

Next we repeated the experiment with typical offset values and found out that frame response times dropped by almost 50% for the lower-priority frames (see "utilization gain" in Figure 4). This improvement results from the load balancing that the offsets introduce.

We determined these typical values based on experience from other projects in which we were able to compare experimental assumptions with real CAN traces. It turned out that the SymTA/S typical values represent a reasonable approximation to the real world. In other words, the results resemble realistic timing profiles of the studied CAN buses, even though detailed design data was not available.

OFFSET OPTIMIZATION

In a final experiment, we used the SymTA/S automatic exploration plug-in [3, 17] to optimize the offsets in order to reduce the response times further by more than 35%. The results of these experiments are summarized in Figure 4. Each curve represents one experiment. The response times (y-axis) are shown for a representative subset of frames in the order of their priority (x-axis). The frame number 23 (by the bold arrows) resembles a response time improvement from initially 90ms to 45ms (with typical offsets) down to 30ms (with optimized offsets).

It might surprise that —without changing the number of frames, their length and CANId, the bus speed or the average bus load— such an enormous improvement in bus utilization was achieved. These experiments show that offsets in particular make a huge difference in bus

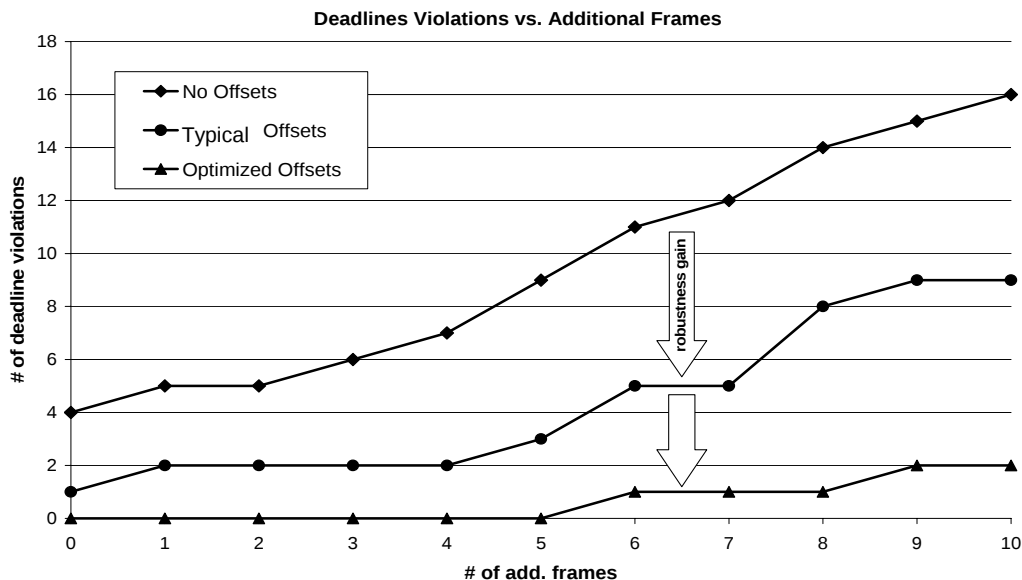


Figure 5 Reducing deadline violations through offset optimization

utilization and responsiveness of frames. These are the important dynamic properties of “performance”.

We conclude that having an analysis that allows comparing and reasoning about offsets systematically, provides new possibilities for OEMs to optimize their networks without the need for changing the design process.

ROBUSTNESS AGAINST ADDITIONAL FRAMES

We proceeded with another set of experiments. This time, we were interested in the *robustness* of each bus configuration against additional frames. We wanted to know how many and what frames can be added to a given bus without violating constraints. To analyze this robustness, we gradually added more and more asynchronous high-priority frames to each configuration. We analyzed the new response times and determined which and how many frames miss their deadlines. These additional frames could be *mixed* or *direct* frames, or they could model frame retransmission that result from bus errors. In principle, the experiment provides a general measure that could be particularized further.

The results are shown in Figure 5. Again, each curve resembles one of the three known configurations. On the x-axis, the number of additional frames is provided. The y-axis captures the number of frames for which the deadline (10% of frame period, at least 40ms) is violated. The optimized configuration can accept significantly more additional frames before a deadline violation occurs. We call such a configuration *robust*, because the network can safely carry more frames without violating any performance requirements. The number of deadline violations in the typical configuration increases much faster. In that case the bus is much more *sensitive* to additional frames, and therefore less extensible.

The curves indicate a quality increase similar to that in Figure 4, but this time in a more sophisticated context. Not only response times are considered, but they are also related to performance requirements/constraints such as deadlines. Scheduling analysis using SymTA/S offers a wide range of such views that can be customized and extended.

DETAILED SENSITIVITY ANALYSIS

In a final set of experiments, we looked closer at the *sensitivity* of individual frames. We again started with a typical offset configuration. Then we varied individual offset values and tracked the influence on individual frame response times. Figure 6, Figure 7, and Figure 8 illustrate the dependency of selected individual offsets to individual frame response times. The sensitivity plug-in for SymTA/S [9] performs such experiments automatically and also produces the figures.

These detailed results emphasize the strong relationship between frame offsets and frame responsiveness,

robustness, and sensitivity. Moreover, they show that controlling those relations offers an enormous optimization potential that can be exploited in practice.

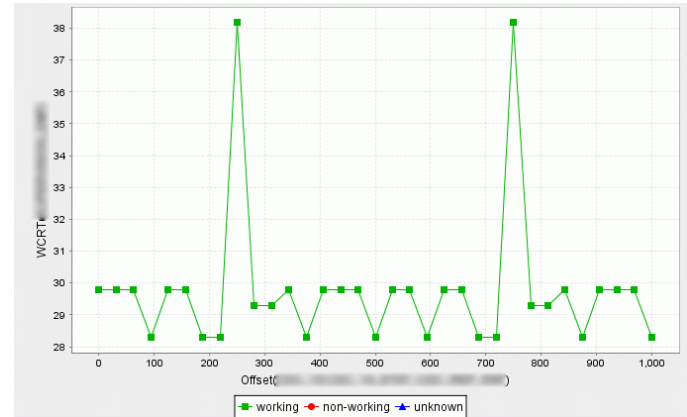


Figure 6 Frame Y is relatively robust against offset X changes; only a few values should be avoided, e.g. 250 or 750, as they increase the response time by 10ms,

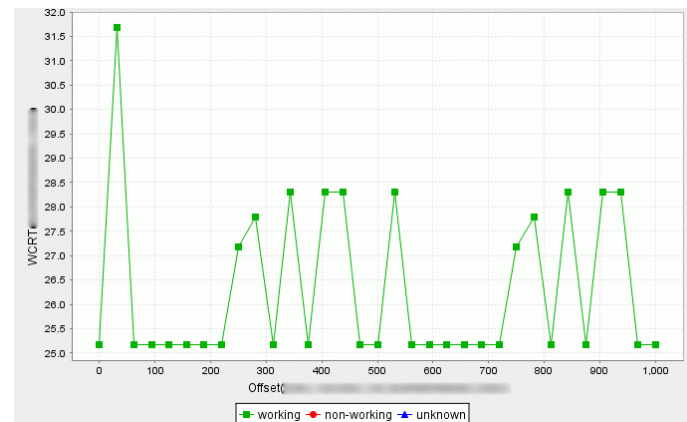


Figure 7 The frame Y response time has a highly irregular dependency on its own offset

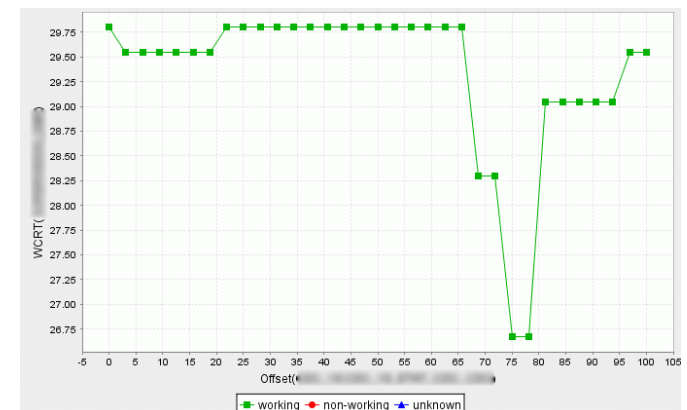


Figure 8 The frame Y response time is very sensitive against offset X changes; only few offset values yield an optimal response time

SUPPLY-CHAIN CONSIDERATIONS

From the experiments, we learn that offsets make a huge difference and can decide about good or poor performances of a bus configuration. Choosing offsets well results in more robust buses and allows adding more frames. This has a number of obvious advantages: it increases reliability, because timing can be effectively controlled. The life-cycle of a network platform increases, since the available bandwidth can be optimally utilized. The risk of late integration surprises is heavily reduced, as key timing dependencies are known.

Interestingly today, no such apparent design improvements have been realized in a broad range, simply because offset information is currently not subject to optimization. Worse, offset information is currently rarely exchanged between suppliers and OEMs. Only the period, length, and CANId are exchanged in form of the so called “communication matrix”.

The good news is that OEMs can use “what-if” analysis, like exploration and sensitivity analysis, to evaluate a huge number of different configurations, including a variety of offset distributions. The experiments in this paper demonstrate that using “typical values” already provides a view quite close to the final system. Other properties, such as the number of mixed or event-driven frames or bus errors, can also be subject to “what-if” analysis, as the experiments with additional frames show.

This way, the critical bottlenecks can be foreseen systematically and extremely early in the design process, way before ECU prototypes are available for test. The technology further enables OEMs immediate reaction in order to eliminate these bottlenecks. There are several options.

Based on the *sensitivity analysis* results, one can derive offset constraints for the most critical (or sensitive) frames as requirements for the ECU suppliers. In contrast to the seeming “data (un)availability problem”, we can thus turn the tables and use analysis to produce data for the supplier, with huge benefits for the supply-chain processes.

As another example, gatewaying strategies, that also affect offset relations, can be optimized. These are usually under the control of the OEMs and provide much more parameters that can be tuned such as queue configuration, which is not shown further here.

The idea to determine requirements and formulate contracts is not new. It is well established for quite some time in hardware IC design, and recent research has considered contract-based design, in particular with respect to timing, also in the area of automotive software engineering processes [2]. Such contracts need concrete requirements, and it is essential that the key requirements are determined early, when the design is

still flexible and offsets can still be changed to optimize the overall design.

Once more, such an approach is only possible with appropriate analysis technology that allows designers to reason about alternatives at the appropriate level of abstraction –like SymTA/S. With traditional methods that require simulators or prototypes, such integration analysis can only be performed very late in the process when optimization and bug-fixing possibilities are very limited.

THE SITUATION OF ECU SUPPLIERS

So far, we have mostly concentrated on the situation of OEMs and what they can do to approach the network integration challenges. Many of them already use the SymTA/S technology. Interestingly, ECU suppliers could benefit from very similar advantages. These shall be briefly examined, even though in practice, ECU suppliers are not yet equally involved in solving system-level timing issues.

First, ECU suppliers always benefit from clear requirements. Having such requirements early when the ECU design is still flexible is of major importance, as late modifications can become extremely time-consuming and expensive for all involved parties.

Secondly, we could –again– turn the tables, as ECU suppliers could also specify requirements on the incoming communication timing. Typical ECU control algorithms rely on new CAN frame data arriving in a determined and timely manner, such that the algorithms always process the most recent numbers. In fact, the frame arrival timing including offsets is a property of the bus, so the OEM is in charge of providing such data. Moreover, arrival timing exhibits jitter due to queuing and bus arbitration. This can be included in the analysis.

We conclude that both OEMs and ECU suppliers can use this technology to a) analyze their system (ECU or bus) based on real data or assumptions, and b) provide the data required by the other party and check if such requirements are met. Besides other components, appropriate analysis methods for CAN as well as operating systems are available as part of our SymTA/S tool chain.

Figure 9 illustrates the duality indicated. For the bus dimensioning, the OEM requires data about ECU2 sending behavior. Likewise, the ECU3 supplier requires data from the OEM. What is initially assumed and required, must later be guaranteed, and vice versa, just as explained in [2].

It will take some time until such a complete process is established in practice, in which OEMs and suppliers flexibly exchange design data in rapid cycles. Since we have already shown how existing technology can improve current design processes significantly, and since this technology is gradually adopted in industrial

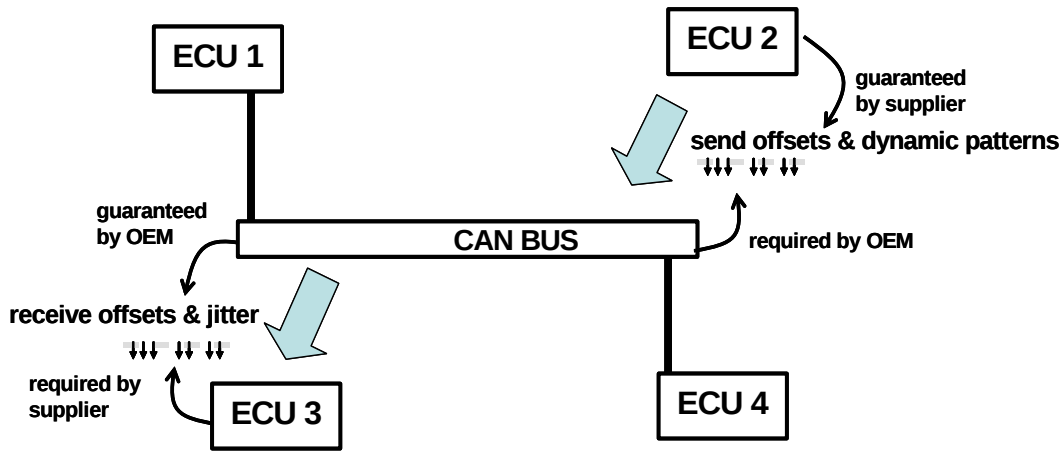


Figure 9 Duality between requirements and guarantees

practice, we will now present a vision of a future supply chain that employs the advanced analysis technology described above.

OUTLOOK: IP PROTECTION & INTERFACES

To develop this vision further, OEMs and ECU suppliers will need to use a common model for exchanging important design information. There are few key requirements on such a model. On the one hand, the model must allow system analysis at a reasonable level of detail and accuracy. On the other hand, the model must allow intellectual property protection of the involved parties when it comes to integration. In other words, it must be possible to specify interface requirements in terms of e.g. data sheets and requirement specifications without disclosing internal implementation details (e.g. ECU task priorities or gatewaying strategies etc.). Balance of interests is crucial since there is no point in modeling and requiring something that cannot be analyzed and verified.

Standardization bodies such as AUTOSAR [1] can help finding and disseminating such common interfaces. The AUTOSAR consortium is aware of this work, but the time for a broad agreement on the “right” AUTOSAR timing model is still to come.

While standardization is inherently slow and must take numerous peculiarities of legacy code into account, SymTA/S already provides a lean but suitable interface model. The models used in SymTA/S are based on event models [11] to explicitly distinguish local component analyses for buses or ECUs from interfaces between these components. The event models capture only basic integration aspects –send/receive frame timing, deadlines, and offsets– and represent an ideal abstraction for the supply-chain communications. For the dynamic properties such as mixed and event driven frames, suppliers can perform ECU analysis and only communicate interface data to the OEM. The same applies to the OEM who performs the network analysis.

OUTLOOK: ITERATIVE REFINEMENT

With such a clear interface, the analysis can finally be repeated as new design details become available, including sensitivity, exploration, and optimization. For instance, OEMs can require suppliers to make more information available. SymTA/S is able to consider offsets of frames and tasks, typically found in the automotive industry, or to consider operating system (OSEK) overhead, complex priority schemes with cooperative and preemptive tasks as well as hardware interrupts. The tool can be adjusted to specific project requirements and system mechanisms. The technical details are, however, not the scope of this paper and can be found in [4, 10].

What is of key importance is the practical possibility to perform the analysis at all in a non-ideal, heterogeneous design process, and based on incomplete specifications. Newly appearing bottlenecks can be discovered quickly and immediate reaction is possible. Secondly and not yet mentioned, freezing certain design parameters can result in new flexibility for other decisions and allows trading the timing reserves and budgets for different components against each other. Third, change requests by suppliers can be tracked.

The experiments illustrate such incremental design. The more information is available, the more accurate the model can be, while OEMs can always answer the question “what is the best we can still reach?” This ensures that, at any given time during the entire development process, the remaining flexibility and optimization potential can be controlled and exploited. The controlled and optimized extension increases network platform life-cycles. In effect, architecture re-designs or the migration to faster buses (e.g. FlexRay) can be safely postponed, maximizing the revenue of one platform. Even in case a re-design can not be avoided, SymTA/S helps in the design of an extensible architecture for the new platform, including FlexRay segments.

CONCLUSION

Recent advances in real-time system theory lead to a flexible, a modular technology that can analyze and optimize complex heterogeneous automotive systems with heterogeneous networks and multi-supplier ECU and software architectures. That technology is incorporated in the tool SymTA/S. Due to its modularity, SymTA/S can handle non-ideal situations, with incomplete or unknown information on one hand, multi-vendor software architectures and legacy components on the other, which are frequently found in practical design scenarios.

The paper presents a case study, which demonstrates how SymTA/S not only provides detailed timing profiles but reveals the design sensitivity to changes and updates, i.e. the design robustness. Examples are given how the sensitivity and robustness of a whole bus or of individual frames against jitters, error, and loss can be easily determined.

There are many more tool features such as system robustness optimization and end-to-end timing optimization which are all based on the powerful modular analysis technology. Several automotive OEMs are already using the tool to optimize and verify their communication networks, starting in very early design stages.

Moreover, SymTA/S has shown to be applicable to emerging heterogeneous networks with bridges and gateways using multiple bus standards. The modular analysis technology allows to include new bus and network standards as they appear while reusing all the previous libraries and design data.

The technology works without disclosure of design details and does not necessarily require new standards and specific design processes. On that basis, we have explained how SymTA/S can be used to improve the interactions along the automotive supply chain leading to better and more robust designs with a minimum invasive effect on the design process. This way, it represents a "stepping stone" to a requirements-driven, system-level design approach.

While that advancement is already feasible with SymTA/S today, the technology could enable a design process innovation in the future. In a parallel research project, we are investigating a tightly integrated "what-if" analysis in the context of formalized design iterations beyond company borders enabling multi-supplier risk management [7].

REFERENCES

1. AUTOSAR Partnership. www.autosar.org
2. J-Y. Brunel, M. Di Natale, A. Ferrari, P. Giusto, L. Lavagno. SoftContract: an Assertion-Based Software Development Process that Enables Design-by-Contract. Proceedings of the conference on Design, Automation and Test in Europe (DATE), 2004
3. A. Hamann, M. Jersak, K. Richter, R. Ernst. A framework for modular analysis and exploration of heterogeneous embedded systems. Real-Time Systems, volume 33, pages 101-137, July 2006.
4. M. Jersak, Compositional Performance Analysis for Complex Embedded Applications, PhD Thesis, Technical University of Braunschweig, Germany, 2004.
5. M. Joseph and P. Pandya. Finding response times in a real-time system. The Computer Journal, 29(5):390–395, 1986.
6. H. Kopetz and G. Gruensteidl. TTP - a time-triggered protocol for fault-tolerant computing. In Proceedings 23rd International Symposium on Fault-Tolerant Computing, pages 524–532, 1993.
7. 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 2005), 2005.
8. C. L. Liu and J. W. Layland. Scheduling algorithms for multiprogramming in a hard real-time environment. Journal of the ACM, 20(1):46–61, 1973.
9. R. Racu, A. Hamann, R. Ernst. A Formal Approach to Multi-Dimensional Sensitivity Analysis of Embedded Real-Time Systems. In Proceedings of the 18th Euromicro Conference on Real-Time Systems (ECRTS), Dresden, Germany, July 2006.
10. K. Richter Compositional Scheduling Analysis Using Standard Event Models – The SymTA/S Approach, PhD Thesis, Technical University of Braunschweig, Germany, 2005.
11. K. Richter and R. Ernst, Event Model Interfaces for Heterogeneous System Analysis, In Proceedings of Design, Automation, and Test in Europe Conference, Paris, France, 2002.
12. M. Spuri. Analysis of deadline scheduled real-time tasks. Technical report, INRIA, Le Chesnay, France, 1996.
13. SymTA/S Project. Institute of Computer and Communication Network Engineering, Technical University of Braunschweig, Germany, www.symta.org
14. K. Tindell. Adding time-offsets to schedulability analysis. Technical Report YCS 221, University of York, 1994.
15. K. Tindell and A. Burns. Guaranteed Message Latencies for Distributed Safety Critical Hard Real-Time Networks. Technical Report YCS 229, Univ. of York, 1994
16. Volcano Network Architect (VNA). http://www.mentor.com/products/vnd/network_design_tools/vna
17. E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report 103, Swiss Federal Institute of Technology Zurich, Switzerland, 2001.