Real-Time Analysis as a Quality Feature: Automotive Use-Cases and Applications

Dr. Kai Richter

Symtavision GmbH
Hans-Sommer-Straße 66
D-38106 Braunschweig
Germany

richter@symtavision.com
Fon (+49) 531 391-3724
Fax (+49) 531 391-3750

Prof. Dr. Rolf Ernst

Institute of Computer Engineering (IDA)
Technical University of Braunschweig
Hans-Sommer-Straße 66
D-38106 Braunschweig
Germany

r.ernst@tu-bs.de
Fon (+49) 531 391-3730
Fax (+49) 531 391-3750

ABSTRACT

Systematic timing and performance analysis allows to quickly detect critical bottlenecks and to estimate flexibility during architecture design, software implementation and system integration. The earlier and more efficiently this is done, the easier it is to choose the best design and implementation alternatives – extensively optimizing the system, thus increasing the quality and profit of a product. It can also improve communication along the supply chain.

Symtavision, a spin-off from the Technical University of Braunschweig, Germany, offers a technology and solutions for real-time analysis that lets system architects and software developers analyze, debug and optimize the system timing in rapid cycles. In this paper we survey a set of interesting use cases that we have studied. Although the mentioned experiments have been performed using our SymTA/S tool suite, we will focus on the applicability and the benefits of timing analysis in general, rather than the SymTA/S technology in particular. We will therefore concentrate on the data required for analysis/optimization, the effects considered, the results obtained, and the practical relevance of each use case.

1. Introduction

Timing and performance problems are an increasing threat to automotive electronics quality and reliability. A major challenge stems from the integration of more and more functions on an increasingly complex architecture. Function integration on an ECU as well as networking of ECUs introduce a multitude of novel dependencies which can be extremely hard to track, let alone predict. For example, what are the minimum and maximum end-to-end delays from a particular sensor signal at ECU A to an actuator response at ECU B, when both ECUs are connected over a communication bus? Will the message always get through? What kind of buffering is needed along the way? Answering these questions is complicated by the established supplier-OEM relationships where no single entity has all required information. Figure 1 contains an example of such a distributed function. An ACC radar or video sensor recognizes a vehicle in the path of the car. Until the car will react and brake, several HW and SW entities are involved incl. ECUs, buses, basic software, drivers, and the application. The overall causality chain is highlighted in the figure. AUTOSAR refers to this as a timing chain. The figure further shows how this timing chain can be decomposed into smaller timing chain segments.

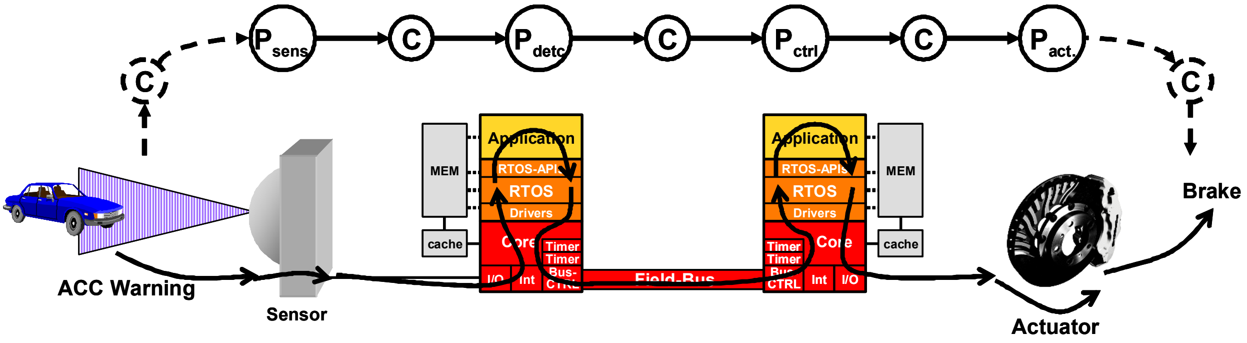


Figure 1 Timing Segments of a Distributed Function

A consumer purchasing an automobile for its electronic functions will to a large extent judge the quality of the vehicle by the performance, responsiveness and reliability of these functions. The perceived quality is at stake if questions such as those posed above are not answered for the many hundred interacting local and distributed functions. Prototyping and test do not efficiently address these issues.

The automotive industry has reacted with a conservative design style. This style has led to cost explosion and inflexible electronics architectures which are reaching the limit of the available space and power in vehicles. Obviously, there is an extreme need for optimization and increased flexibility. Clearly, the largest optimization potential lies in the early design stages, when fewer decisions have been fixed and changes are relatively cheap. The challenge is to make good decisions based on incomplete information.

We believe that system timing and performance are ideal candidates for early analysis and optimization. Furthermore, the initial results can be continuously refined as implementation progresses, thus seamlessly transforming early-stage estimation and optimization into late-stage verification. Figure 2 shows this in a V-Model augmented by timing and performance analysis, flexibility analysis and optimization.

A key enabling property of the approach proposed in this paper is a strict separation of concerns: The techniques described are used *exclusively* to analyze and optimize timing and performance. They do not test *functional* correctness of the system. This allows a tremendous amount of abstraction. The resulting models can be produced quickly and with low effort and explicitly take into account uncertainty and lack of detailed information. The techniques described perform extremely fast compared to simulation, prototyping or test. Furthermore, they focus on the most critical cases, thus avoiding testing the hundreds of thousands of cases that perform without problem. Finally, as indicated in Figure 2 the techniques do not interrupt established design flows. They are optional add-ons that are valuable in any design stage, albeit the largest value is reaped from systematic application throughout the complete design.

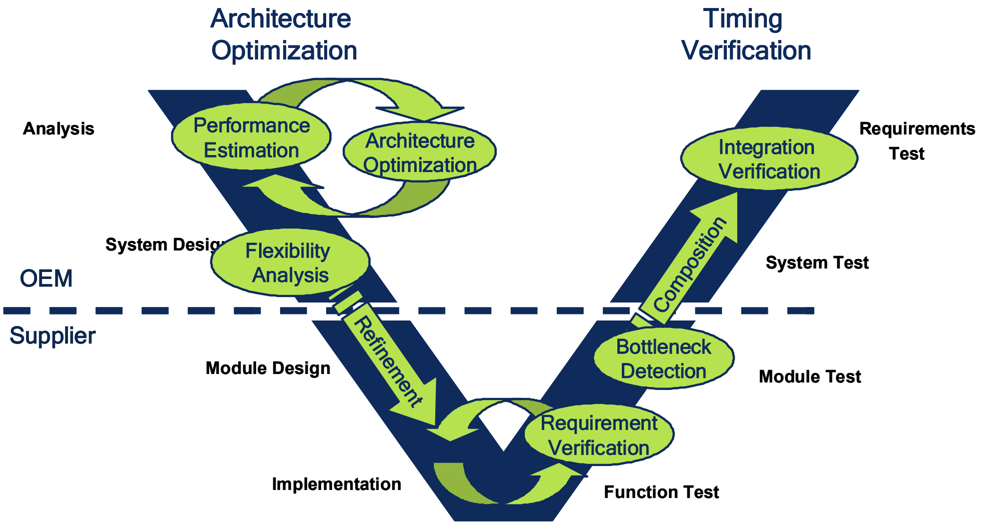
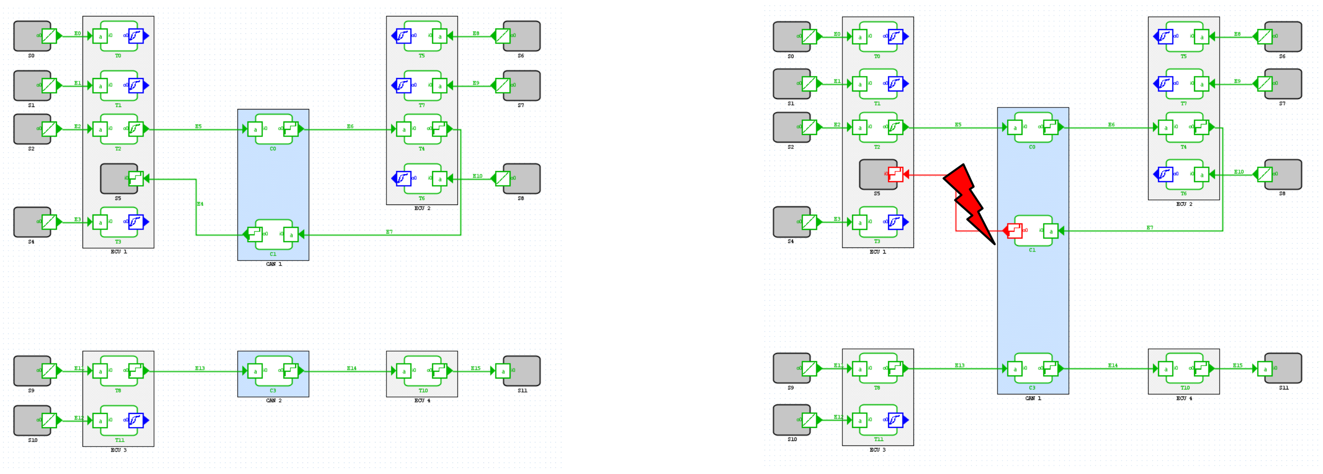


Figure 2 Application of Performance Analysis and Timing Verification

2. Motivating Example



a) Two Independent Subsystems **b) After System Integration**

Figure 3 Example of System Integration Using a Shared Bus

Figure 3 shows a SymTA/S model of two electronic subsystems. The two subsystems are functionally independent and are designed separately by two different electronics suppliers. The automotive OEM would like to integrate both subsystems in a vehicle.

Figure 3 a) shows the system before integration. Both the upper and the lower subsystem are implemented on 2 ECUs (vertical square boxes on the left and right) that exchange messages over a CAN bus (vertical square boxes in the middle). The white, rounded boxes represent software-functions on the ECUs and messages on the CAN-busses, respectively. The gray, rounded boxes model the activation of software-functions (in this case, timers). The gray, rounded box on the upper left ECU is a special case: it constrains the valid *jitter* (i.e. the deviation of timing from a perfectly periodic scheme) of messages arriving via the upper CAN bus.

Timing and performance of both subsystems have been analyzed by SymTA/S. All elements are green, indicating satisfaction of all timing and performance constraints.

Figure 3 b) shows the system after integration. All messages are now transmitted over a single CAN bus, leading to additional load and potentially longer blocking of low-priority messages. After analysis with SymTA/S, it is determined that the aforementioned *jitter constraint* is violated (indicated by the ‘lightening’).

Systematic timing and performance analysis allows to quickly identifying such integration problems. Furthermore, using early-phase optimization, it is possible to find system configurations that will avoid these problems in the first place. For example, the problem uncovered in Figure 3 b) can be resolved by changing some message priorities (CAN IDs), communication buffering strategies or even task and OS configurations. Sensitivity analysis additionally allows to determine early on the available headroom, e.g. for slower ECUs, more complex software-functions, more irregular communication, faster execution rates etc.

In the following sections, timing and performance analysis of our example system is described in detail. Special emphasis will be on

- Typical real-world properties of ECUs and buses that need to be addressed for accurate analysis
- Keeping in mind the separation of concerns between ECU / subsystem suppliers and integrators / OEMs

The latter requires the specification of timing interfaces between suppliers and integrators, in particular

- the specification of jitter requirements and assertions
- the breaking down of end-to-end latencies into timing chains, which are further split into segments.

Upcoming standards for function integration, in particular AUTOSAR, must systematically incorporate timing and performance aspects. Otherwise, these standards will not live up to the promise of simple *and* reliable integration of functions from different sources on the same ECU, remapping of a function from one ECU to another and so on. This paper thus also indicates a path to achieving that goal.

3. Single ECU Scheduling Analysis

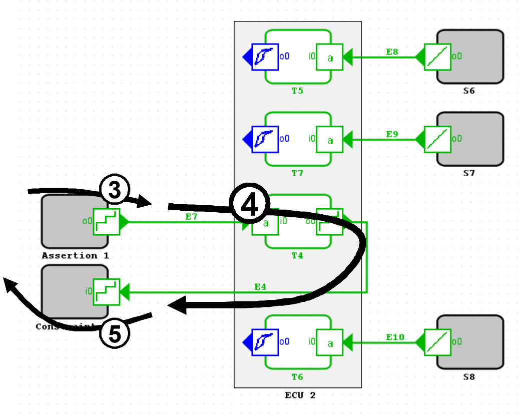


Figure 4 Scope of ECU Supplier, Timing Chain Segments

The ECU suppliers can perform scheduling analysis on their ECUs to determine the response times of tasks and functions that represent an important timing segment within end-to-end latencies. Figure 4 shows part of the entire system from Figure 3. Timing chain segments 3 and 5 are under control of the OEM, while segment 4 can be solely determined by the ECU supplier. Furthermore, the ECU supplier can determine the response message's timing that influences the bus scheduling on segment 5.

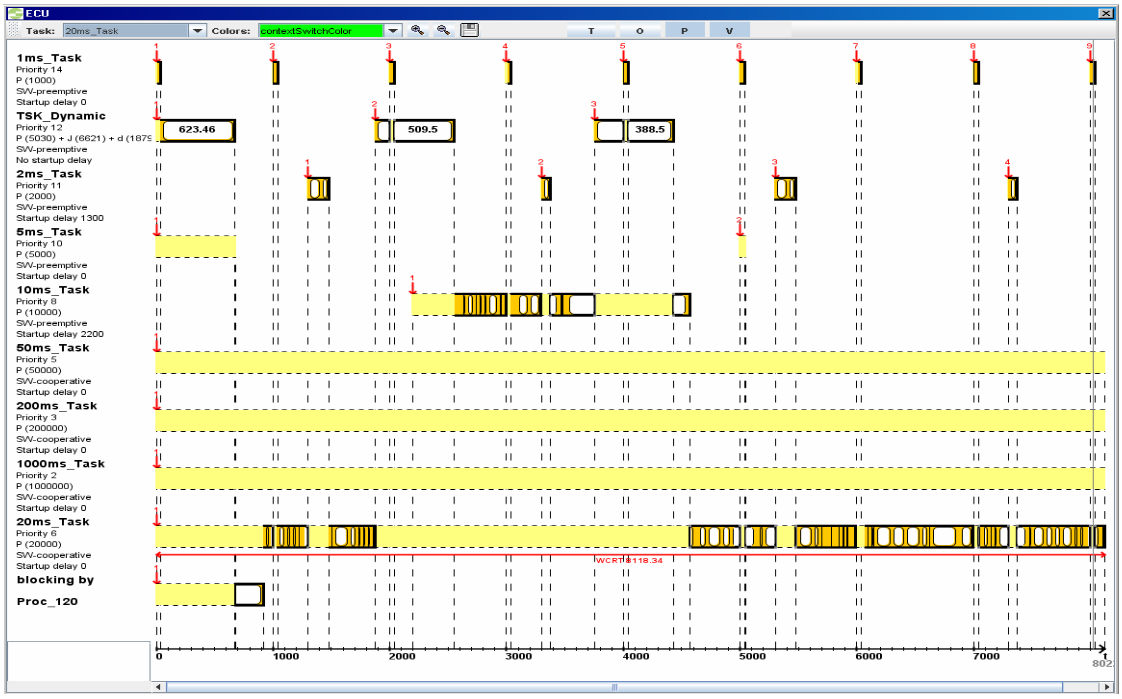
Performing the ECU scheduling requires the following information:

- Operating system scheduler (typically some OSEK implementation)
- For all tasks / processes mapped on that ECU
  - Scheduling parameters (priorities, preemtable / non-preemptable, …)
  - Core execution times, i.e. their "net runtime" *without* interrupts
  - Activation rules and timing, typically time-table configuration, activation chains, worst-case interrupt frequency, …

The ECU configuration can be taken from an OIL file, while the execution times are typically available from measurements during ECU design. Many popular debugging and trace tools use easily accessible xml or csv file formats, which simplifies automatic import functions as found in SymTA/S.

We have analyzed complete ECUs with several hundred processes, thereby considering the specialties of the ERCOSek - RTOS such as TimeTable activation, cooperative and preemptive tasks, asynchronous tasks, OS overhead, interrupts and drivers. We have determined system load, task and process response times, and peripheral access jitters, so that we could compare these against given deadlines and other constraints.

The analyses do not only calculate values, they also provide information under which circumstances certain worst-case situations occur. By this, designers can re-produce these cases in their test system, which further increases designers' confidence and the general acceptance of scheduling analysis. Additionally, the understanding can be considerably enhanced with the help of worst-case scheduling diagrams as depicted in Figure 5. SymTA/S generates such diagrams on demand.



ECU

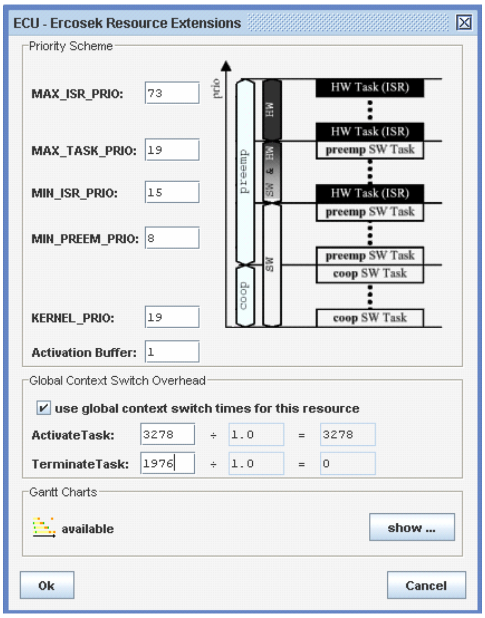
Task: 20ms_Task

Colors: contextSwitchColor

Figure 5 Diagram of Complex ERCOSek Schedule

Unfortunately, scheduling analysis is often wrongfully rated too simplistic, too coarse-grain, and too inaccurate. This is often due to overly simplistic models such as RMA/DMA (rate- / deadline- monotonic scheduling) that do not account for the specific details of real-world designs.

In contrast, we could show that it is in fact possible to adjust the analysis to details of the operating system, or to a complex application behavior. We have successfully deployed other extensions, too, such as alternating task configurations at run time, the distinction of several AppModes, and the consideration of burst interrupts, as shown in Figure 7. SymTA/S and the ERCOSek analysis library further consider the influence of preemptive and cooperative tasks, TimeTable activations, OS overhead, etc. Such extensions lead to an analysis accuracy of 95% and more compared to the measured values, which makes this kind of analysis a serious and promising building block in the design process.



ECU - Ercosek Resource Extensions

Priority Scheme

MAX_ISR_PRIO: 73

MAX_TASK_PRIO: 19

MIN_ISR_PRIO: 15

MIN_PREEM_PRIO: 8

KERNEL_PRIO: 19

Activation Buffer: 1

Global Context Switch Overhead

☑ use global context switch times for this resource

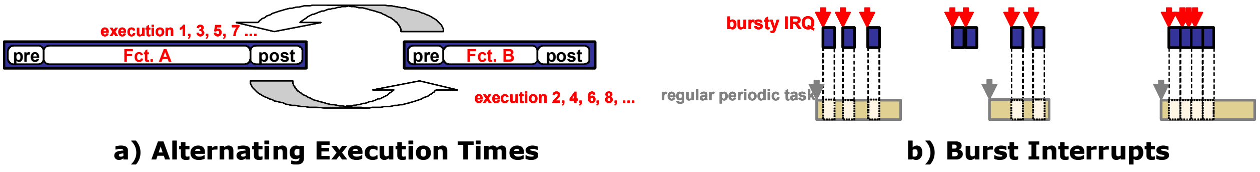
ActivateTask: 3278 ÷ 1.0 = 3278

TerminateTask: 1976 ÷ 1.0 = 0

Gantt Charts

available

Figure 6 ERCOSek Priority and OS Configuration in SymTA/S



a) Alternating Execution Times

b) Burst Interrupts

Figure 7 Extensions to the ERCOSek Operating System Analysis

4. Extensions to Single ECU Analysis

It is further possible to analyze other dynamic influences such as RPM-dependent task frequencies; the higher the engine RPM, the more often the ignition task must be executed, and it must be in synch with the crank shaft. Since analysis is so fast, it can be performed for a characteristic RPM-scheduling curve within minutes, without any additional testing or measurement effort. Clearly, such curve traversal can be automated to further reduce user interaction.

Such "what-if" analysis also enables highly efficient optimizations, as it allows configuration parameters to be varied without re-building, re-testing, or re-measuring the system. SymTA/S, specifically, contains a plug-in for automatic elimination of deadline violations, reduction of controller dead time, or jitter minimization for accessing peripherals. The user controls which configuration parameters can be changed and which are fixed. We have exploited this possibility in a dual-processor ECU design to reduce the jitter of the inter-processor communication and to make the entire system more stable.

As a third general application for such "what-if" analysis, one can vary the execution times or task periods to determine the so called "criticality" of a particular process or task. By this, the designer of a software process knows how much head room there is left for later modifications along a particular timing segment, or by how much the time budget must be reduced to meet a local constraint. This quickly uncovers upcoming bottlenecks as well as hidden flexibility, and allows optimizing the system further. This has helped us in finding optimized function distributions in a dual-processor ECU, as well as in robustness optimization of CAN busses.

5. CAN Bus Analysis

OEMs and other bus integrators can perform scheduling analyses to determine the message timing, i.e. the timing chain segments 3 and 5. These are an integral part of the aforementioned inter-ECU timing chain. Furthermore, bus analysis allows reasoning about message send and receive jitters, and such information can be modeled using the constraint/requirement model, and communicated along the supply chain to those ECU suppliers that are responsible for timing chains 2, 4, and 6. And we already know who these are.

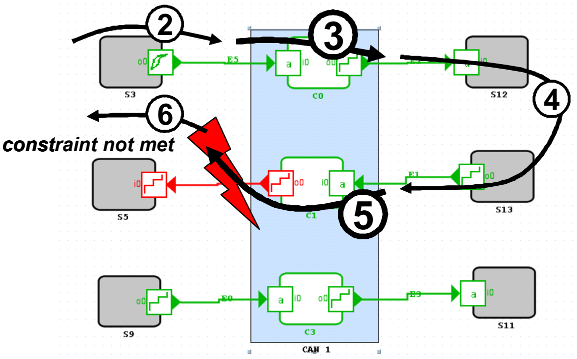


Figure 8 Isolated Bus View with Assertions and Constraints

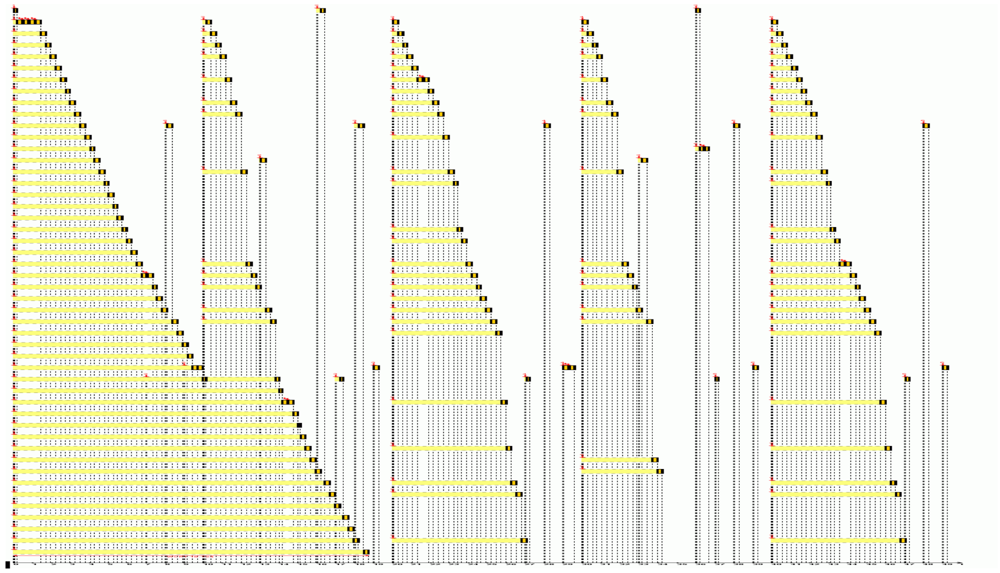


Figure 9 Scheduling Diagram of Complex Bus with Jitters and Bus Errors

Basically, bus analysis is similar to ECU analysis. Instead of tasks, messages need to be scheduled. For CAN buses, a so called "K-Matrix" provides key bus configuration information incl. message IDs, length, and period.

While K-Matrices allow designers to perform a "first shot" of analysis to obtain the average network load, more information is needed to provide more detailed results. Message jitters —at sender and receiver side— strongly influence the bus behavior, but they are not part of the K-Matrix description. Again, the possibility of "what-if" analyses allows reasoning about the effects of jitters using intuitive assumptions.

In own experiments, we started by assuming message deadlines equal to the periods with zero send jitters and then varied the jitters within a typical range of 10-30% of the message's periods. This increases the non-determinism in the message timing, increases the dynamic peak-load, and might lead to transient overload situations. For each jitter distribution we detected those messages that will miss their deadlines and those that could be lost in the worst case. We performed these analyses within seconds without any test, simulation, or measurement.

The accuracy of these analyses can be increased even further as more detailed information about the message timing is taken into account. Usually, TimeTable-driven COM-tasks at each ECU send out messages with a certain known phase delay or offset. This produces "gaps" in the communication pattern of each ECU. The sharing of message objects (MOs) rules out certain bus conflicts on one hand, while introducing blocking of higher-priority messages through lower-priority messages residing in the shared MO. Additionally, it is possible to analyze the effect of bus errors and retransmissions. Using our approach it becomes easy to show that some messages are relatively sensitive to jitters and errors with respect to loss and deadline verification, while others are quite "robust". It is also easy to show how many additional messages the bus could possibly carry, e.g. for diagnostics and flashing, before any other deadline was missed.

The CAN analysis library, that Symtavision offers, provides users very flexible automatic optimization capabilities. In one project, we have configured the optimizer to favor robust configurations over sensitive ones, resulting in a system that can tolerate significantly more transmission errors than the original one. Figure 9 shows a minified scheduling diagram of a system with more than 50 messages with jitter and bus errors. Another important influence is the additional delay due to gatewaying. An appropriate gateway model is currently under development, in particular for heterogeneous, e.g. CAN / FlexRay, networks.

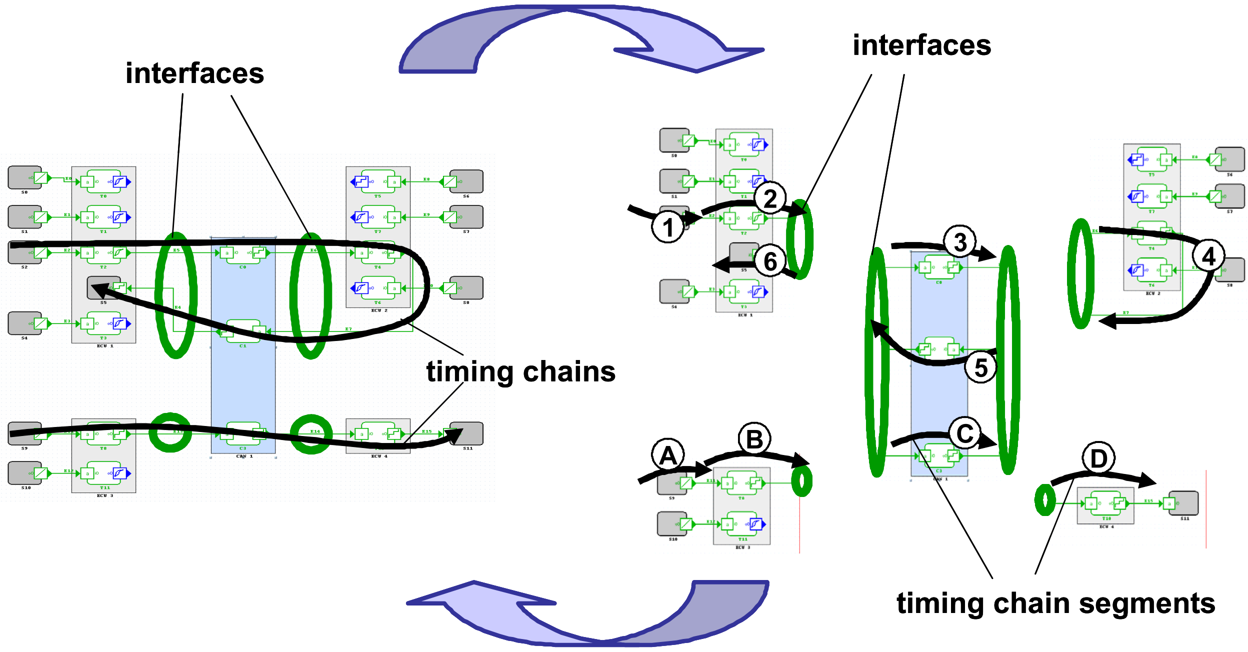


Figure 10 Timing and Performance Interfaces in Supply Chain Communications

6. Supply Chain Communication

A question of central concern with network analysis is "where does the information about message jitters and TimeTables come from?" We have mentioned that the established K-Matrix focuses on average values such as periods but ignores jitters and message offsets stemming from TimeTable COM-tasks. Such information is generated, however, as a side effect during ECU analysis that we have surveyed. In other words, the information can actually be produced using real-time analysis. Why not communicate it to the bus integrator? Likewise, the OEM can provide its suppliers with information about the dynamic behavior of CAN messages, needed for considering receive interrupt and driver behavior in ECU scheduling analysis. When concentrating on the key timing information, such data can in fact be communicated among the supply chain without disclosing any intellectual property of implementation details. Figure 10 illustrates the interfaces that OEMs and suppliers can use, and how timing chains split up into several segments with known responsibility.

With respect to the V-model of Figure 2, this improves the bottleneck detection and integration analysis on the verification path. There is, however, another, possibly even more beneficial application of such analysis in the early phases of architecture optimization and flexibility analysis. The experiments from Section 5 show that, even when no detailed information is available yet, "what-if" analyses lets OEMs explore and analyze a huge number of possibilities including a variety of jitter distributions, different error models, and many more not being presented further here. This way, the critical bottlenecks can be *foreseen* systematically and extremely early in the design process, way before ECU prototypes are available for test. This allows constraints for the most critical (or sensitive) messages to be formulated as requirements for ECU suppliers. The same holds for ECU suppliers, as they might also have requirements on the incoming data and benefit from known freedom for optimization. Roughly speaking, "missing data" need not necessarily be considered a problem for analysis. Rather, it represents design flexibility that can be exploited and controlled. It allows specifications and requirements to be formulated and refined extremely early and with huge benefits for the supply-chain processes.

It is essential that the key requirements are determined early, when the design is still flexible and, for instance, CAN message IDs can still be changed to optimize the overall design. As another example, gatewaying strategies can be optimized. These are usually under the control of the OEMs and provide many parameters that can be tuned such as queue configuration, which is not shown further here.

Once more, such procedure is only possible when appropriate abstract models are available early. 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 as late modifications can become extremely time-consuming and expensive. Thus, providing such information early significantly increases the analysis and optimization possibilities of all involved parties.

The result would be a detailed and reliable network analysis based on black-box ECU descriptions. We forecast that the availability of timing data will be an important requirement of OEMs when selecting suppliers. Furthermore, the ability to perform "what-if" analysis in rapid cycles enables a multi-supplier risk-management that allows reacting to bottlenecks earlier than ever and in line with the projected road map. Moreover, the Institute of Computer and Communication Network Engineering is currently researching new risk-management systems by applying sophisticated reward/ penalty models for over-/under-satisfying requirements based on "what-if" analysis.

7. Conclusion

We have briefly surveyed several applications of real-time analysis in the automotive area. All examples carefully considered real-world design and business processes with respect to data availability, relevant results, etc.. As a key conclusion, we see that the availability of timing information often makes a huge difference. ECU suppliers can provide all the necessary info while protecting the essential IP. The other way round, the OEMs can use "what-if" analysis, e.g. analyze the dynamic behavior of messages, and formulate requirements for the suppliers using the same "language".

The case studies did not rely on any simulation, nor did they require prototypes nor test equipment. Quite to the contrary, OEMs can evaluate different network choices upfront and use a technology like SymTA/S to dimension optimized and robust buses with known extensibility. Likewise, ECU suppliers can explore and control their flexibility. Understanding, analyzing, and specifying these properties early, way before prototypes are available, reduces the risk of integration on both sides. And it allows timing chain segments to be analyzed individually, and integrated.

We forecast that the availability of such information will be an important requirement for OEMs when selecting suppliers. This provides a new quality in the system-level integration process, as component integration along the supply-chain becomes more reliable and more systematic than ever before. Upcoming standardization approaches such as AUTOSAR will ultimately need to cope with these questions. The technology is available already now.

SYMTA VISION

Symtavision GmbH
Hans-Sommer-Straße 66
38106 Braunschweig, Germany

Tel: +49 531 391-3729
Fax: +49 531 391-3750

info@symtavision.com
www.symtavision.com