How OEMs and Suppliers can face the Network Integration Challenges

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

1: Dr. Kai Richter
Symtavision GmbH
Hans-Sommer-Straße 66
38106 Braunschweig, Germany
richter@symtavision.com

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

Abstract

Systems integration is a major challenge in many industries. Systematic analysis of the complex integration effects, especially with respect to timing and performance, significantly improves the design process, enables optimizations, and increases the quality and profit of a product. And it helps to improve supply-chain communications. This paper surveys a set of interesting experiments we have conducted on a real-world automotive communication network using our new SymTA/S system-level schedulability analysis technology. We demonstrate that, and how, analysis technology helps answering key integration questions, thereby carefully respecting the established business models.

1 Introduction

The increasing application complexity, together with a strong time-to-market pressure, require 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 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, incl. transient overload, buffer under- and over-flows, and missed deadlines. All electronics industries have identified this as a major bottleneck in the design, and have started standardization efforts for components, e.g. AUTOSAR [1] in the automotive area.

We have been researching the role of performance and timing analysis in system-level design in our SymTA/S project [2] for more than five years. We have adopted and extended a host of theoretical contributions to meet industrial requirements, and we have successfully demonstrated our technology in industrial projects. In this paper, we will summarize the application SymTA/S (Symbolic Timing Analyses 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, thereby carefully respecting the established business models along the supply-chain.

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

2 The Network Integration Problem

Network integration is carried out by an OEM who determines bus topology, speed, number of nodes and messages, and finally the configuration, e.g. the assignment of priorities or time slots to bus messages. The decision making process typically includes the following questions: Is the network (temporarily) overloaded? Which messages can get lost, and how often? Can more ECUs (electronic control units) and how many be connected without overloading the bus? How about diagnosis and ECU flashing? Answering these questions requires understanding the sophisticated effects that individual decisions might have on the overall performance and timing. And 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 is established common practice but suffers from serious corner case coverage problems and is not suitable to reliably detect the bottlenecks. Therefore, architects very often favor less efficient, conservative designs. For instance, conservatively allowing "N out of M" messages to get lost is not an unusual way to “guarantee” that a minimum number of messages gets through. But sending significantly more messages than actually "required" further increases bus load and should be avoided, since this also increases the number of lost messages. Detecting and reducing such inefficiencies, in turn, requires knowing how message loss can be reliably analyzed and bounded. The lack of systematic procedure currently prevents OEMs from thorough optimizations, and overly conservative approaches are common practice.

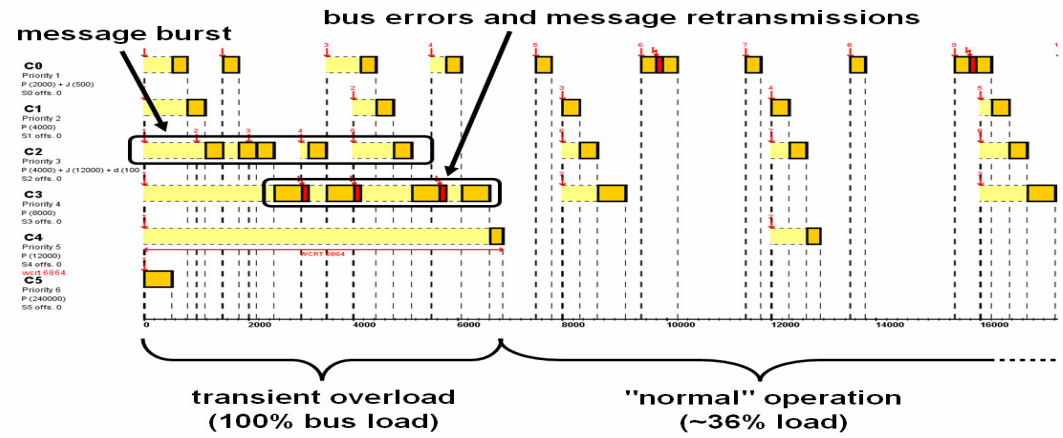

Figure 2 Message Jitters, Burst, and Errors Result in Complex Communication Patterns

3 Quick Review

3.1 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 message, multiply the frequency of a message (1/period) with its length (incl. protocol overhead), build the sum over all messages, and finally divide it by the network bandwidth. The result is a relative network load, often called utilization, given in percent of the available bandwidth.

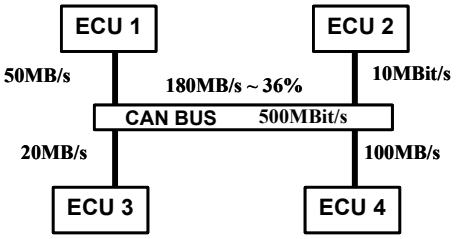

Figure 1 Simple Load Analysis Example

Figure 1 shows an example. Four ECUs produce bus traffic that accumulates on the shared bus. The resulting total traffic is 180kBit/s which represents a load of 36% 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; some say 40%, others say 60%. Why is that? Clearly, increasing the load means better utilizing resources which translates into promising cost savings. The load model says, however, nothing about if deadlines can be met or buffers overflow and should, therefore, be used with care.

In fact, 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.

3.2 Dynamic Communication Patterns

Finding bottlenecks requires the complex dynamic communication patterns to by analyzed. A host of sophisticated methods from real-time scheduling theory is available to determine and analyze such patterns automatically [3,4,5,6]. In this paper, we will not introduce the theoretical foundations of this work. We rather emphasize the practical impact of having such techniques.

The methods directly provide detailed data such as response times that allow answering key design questions. For instance, to guarantee that a message X will never get lost (overwritten in the sender's buffer), its maximum response time must not exceed its minimum re-arrival time (the deadline, often the period). Calculating the response times requires consideration of the protocol-specific behavior of the CAN bus. Higher-priority messages can significantly delay lower-priority ones, message jitters further distort the timing behavior, the controller type (basicCAN, fullCAN, etc.) influences the order in which messages are sent, and bus errors can lead to additional retransmissions.

Figure 2 shows such a complex communication sequence with a variety of influences. Key of such analysis techniques is that they find and evaluate the critical situations automatically without user interaction, provided that the system configuration, i.e. the message IDs (priorities), message length, jitters, the interface queues, and an error model is known. Figure 3 structures this required information into bus-related, ECU-related, and models for error and flashing/diagnosis. Once this data is available, one can select among a host of more or less powerful schedulability analysis techniques, some are available as tools.

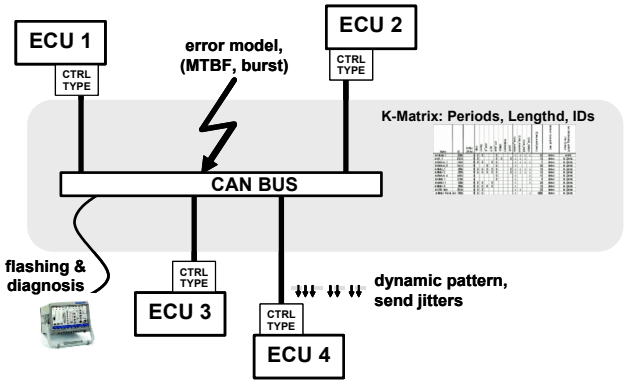

Figure 3 Information Required for Reliable Schedulability Analysis

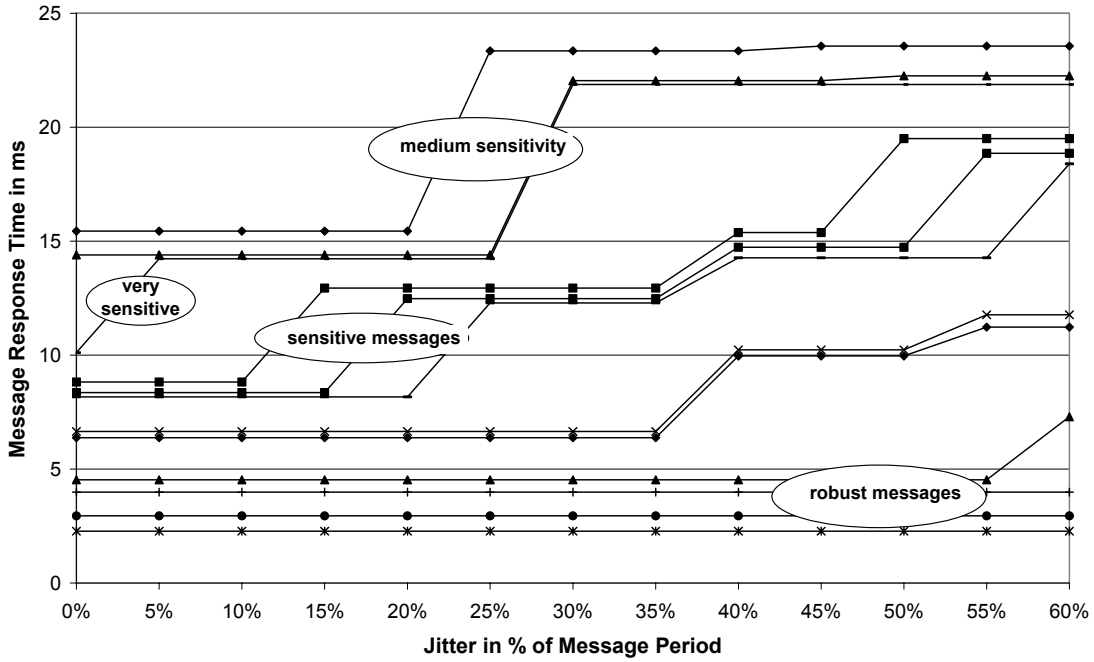

Figure 4 Jitter-Sensitive and Robust Messages illustrated by Response Time/Jitter Dependency

3.3 Schedulability Analysis in Practice

In practice, however, only part of this data is available to the OEM, usually the CAN K-Matrix, covering only the static part of the system. The gray area of Figure 3 illustrates the "scope" of OEM information. Those important dynamic influences, such as message send jitters, result from ECUs implementation decisions and are often not available in detail. At best, the used CAN controller type is known early. Errors and flashing even have their roots outside the actual design task with little information. So, can such analysis technology really help when data availability is a major concern? In fact, it can, as we will see by the end of the next section.

4 The Case Study

We have used our SymTA/S tool suite to analyze a real-world power train CAN bus from the automotive industry. Several ECUs (electronic control units) including gateways are attached to that bus, each sending and receiving a total number of more than 50 messages. We automatically imported the length, CAN id (priority), and the period of each message from the K-Matrix. We knew the jitters of only a few messages, typically in the range of 10-30% of the message's period. Other jitters were unknown, as the ECU implementations were not yet completed. Using our tool suite, we conducted a set of experiments, each based on different assumptions on the missing information.

Simulators or prototypes were not required as SymTA/S uses analyses known from real-time scheduling theory that require only few key parameters that are usually known or can be estimated. Such models can also be tailored to specific real-world mechanisms of protocols (and OSes) and yield quite high accuracy. Finally, the forecasted results can be validated in the final system, although this requires extensive testing.

In the first experiment, we assumed zero jitters and verified that all messages will meet their deadlines. In fact, such simplifications (zero jitters, no errors) have a limited practical relevance. Very important is ,however, the fact that we could do such "what-if" observations within minutes, without any simulation, prototype or test equipment.

We proceeded with other experiments, in which we assumed "realistic" jitters for the unknown messages. We also considered different types of bus error models that lead to retransmissions. Practically useful errors models are available for sporadic errors [7] that appear once in a given time interval (similar to the MTBF –mean time between failures– model) and for burst errors [8].

4.1 Sensitivity and Robustness

We repeated these experiments, and within minutes we determined how message response times vary over several jitter and error distributions. We found out that some messages are relatively sensitive [9] to jitters and errors, while others are quite "robust".

Figure 4 illustrates the dependency between the jitter and the response time for few selected messages. A message whose response time increases fast with increasing jitter is considered sensitive, messages with relatively constant response times are considered robust against jitters. Similar results have been obtained for error-sensitivity.

4.2 Message Loss

We further detected, for each experiment, how many and which messages already miss their deadlines and could be lost. The dotted lines in Figure 5 illustrate a selected part of the results. The x-axis captures the increasing jitter (in %), while the y-axis shows the number of messages (in % of all messages in the K-Matrix) that will miss their deadlines. When ignoring bus errors (best-case line), no message gets lost until the jitters exceed 25% of the message periods, then loss is slightly increasing. In the worst case experiment we considered burst bus errors, bit stuffing, and the minimum re-arrival time as a deadline. We can observe deadline violations and message loss starting at very small jitters and increasing rapidly, an undesired behavior.

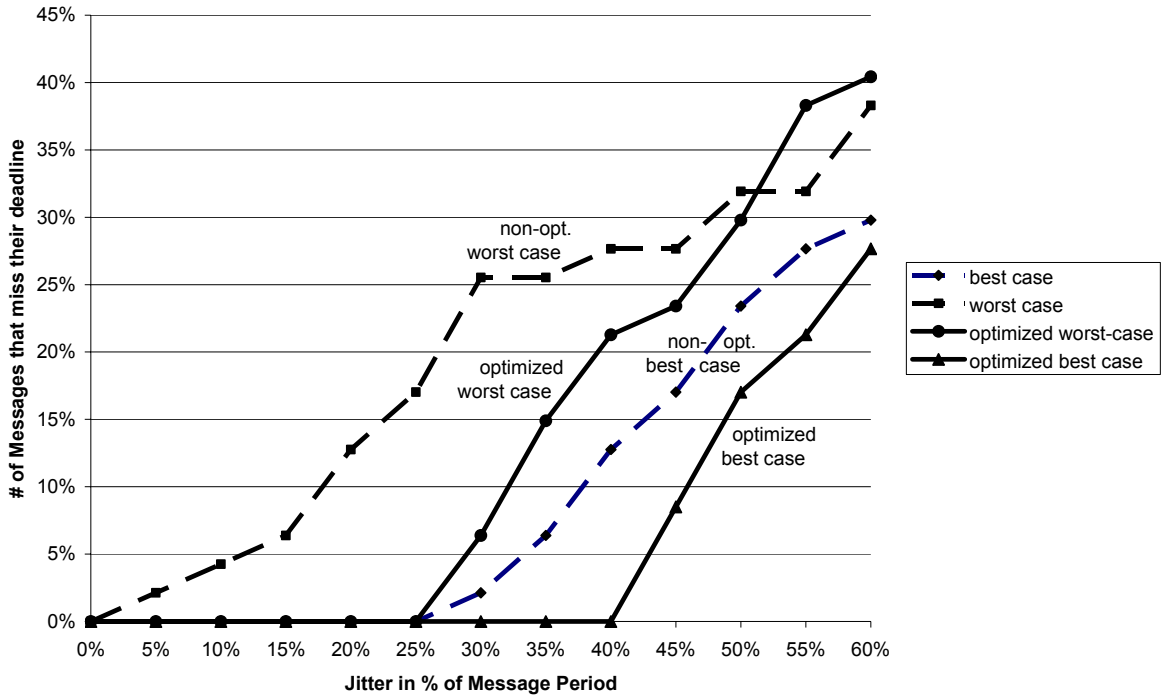

Figure 5 Message Loss at Different Jitter Values Before and After Bus Optimization

4.3 CAN Optimization

In order to eliminate this message loss we were looking for optimized priority (CAN ID) configurations. We used the automatic optimization feature of SymTA/S to find better CAN ID configurations that would exhibit less message loss. The optimizer [10] also performs "what-if" analysis using genetic algorithms. We configured the optimizer to favor robust configurations over sensitive ones. Quickly, we obtained a system that does not loose a single message at 25% jitter, even in the presence of errors and bit stuffing. Figure 5 allows comparison between the optimized (solid lines) and original (dotted lines) system.

5 Supply-Chain Considerations

These experiments show that, even when no detailed information is available, "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. The technology further enables OEMs immediate reaction in order to eliminate these bottlenecks. There are several options.

For instance, once the "sensitivity analysis" of Section 4.1 has been conducted, jitter constraints for the most critical (or sensitive) messages can be formulated as requirements for ECU suppliers. In contrast to the seeming "data (un)availability problem" as it was mentioned in Section 3.3, we can obviously turn the tables and use analysis to produce data in the other direction 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. 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.

5.1 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. Interestingly, very similar advantages apply to the suppliers, too. These shall be quickly surveyed.

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

Furthermore, we can –again– turn the tables as ECU suppliers also have requirements on the incoming communication timing. Typical ECU control algorithms, designed using e.g. MATLAB/Simulink, rely on new CAN message data arriving in a dedicated timely manner, such that the algorithms always process the most recent numbers. In fact, the message arrival timing including the jitter is a property of the bus, so the OEM is in charge of providing such data.

Without further demonstration, we summarize 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 tool chain.

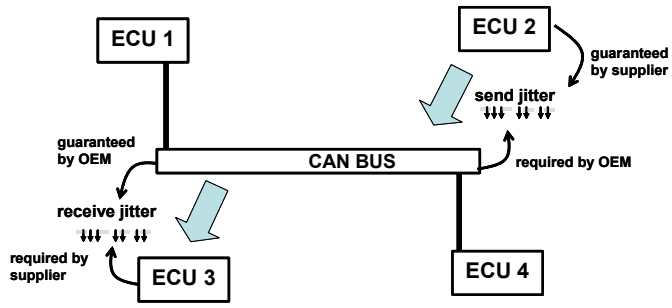

Figure 6 Duality of Requirements and Guarantees

Figure 6 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.

5.2 IP Protection & Interfaces

For this duality, OEMs and ECU suppliers must 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 analyzing the system at a reasonable level of detail and accuracy. On the other hand, the model must allow protecting the intellectual property 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.).

The model used in SymTA/S uses event models [11] to explicitly distinguish local component analyses for buses or ECUs from interfaces between these components. The event models capture only key integration aspects such as send/receive message jitters, deadlines, etc.. and ignores details of ECUs and buses, and therefore represent an ideal abstraction for the supply-chain communications. Suppliers can perform ECU analysis and only communicate interface data to the OEM. The same applies to the OEM who performs the network analysis.

5.3 Iterative Refinement

With such a clear interface, the analysis can be repeated as new design details become available, incl. sensitivity, exploration, and optimization. SymTA/S is able to consider TimeTable activation of messages and tasks, typically found in the automotive industry, considers operating system (OSEK) overhead, complex priority schemes with cooperative and preemptive tasks as well as hardware interrupts, and can be adjusted to specific project requirements and system mechanisms. The technical details are, however, not in the scope of this paper and can be found in [12,13].

What is key is the practical possibility to perform the analysis at all. Newly appearing bottlenecks can be discovered quickly and immediate reaction is possible. Secondly and not yet mentioned, freezing certain design parameters can result is new flexibility for other decisions and allows trading the timing reserves and budgets for different components against each other. This ensures that, at any given time during the entire development process, the remaining flexibility and optimization potential can be controlled and exploited.

6 Conclusion

From these and other examples that we have studied, we conclude that the availability of timing information and the capability to exploit it often makes a huge difference. In fact, ECU suppliers can perform analysis and provide all the necessary info, at the same time protecting their essential IP by using clear interfaces to communicate key timing information. The other way round, the OEMs can use "what-if" analysis and formulate requirements for the suppliers using the same "language".

The experiments 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.

While the examples in this paper have merely concentrated on single interfaces between two parties, one OEM and one ECU supplier, buses typically connect several ECUs and can further form heterogeneous networks with bridges and gateways. From a system-level perspective, this leads to heterogeneous networks of distributed black-box components that must also be analyzed including the complex dynamic dependencies between all components. Once more, the availability of interface information will be a key enabler for such system-level considerations.

We forecast that the availability of such information will be an important requirement for OEMs when selecting suppliers. The ability to perform "what-if" analysis in rapid cycles could also enable a multi-supplier risk-management [14], possibly in combination with a penalty-reward model, that allows reacting to bottlenecks earlier than ever and in line with the projected road map.

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. The technology is available already now.

References

[1] AUTOSAR Partnership. www.autosar.org

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

[3] C. L. Liu and James W. Layland. *Scheduling algorithms for multiprogramming in a hard-real-time environment*. Journal of the ACM, 20(1):46–61, 1973.

[4] M. Joseph and P. Pandya. *Finding response times in a real-time system*. The Computer Journal, 29(5):390–395, 1986.

[5] 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.

[6] M. Spuri. *Analysis of deadline scheduled real-time tasks*. Technical report, INRIA, Le Chesnay, France, 1996.

[7] K. Tindell and A. Burns. *Guaranteed message latencies for distributed safety-critical hard real-time control networks*. Technical Report YCS 229, Department of Computer Science, University of York, UK, May 1994.

[8] S. Punnekkat, H. Hansson, and C. Norström. *Response time analysis under errors for CAN*. In Proceedings of the 6th Real-Time Technology and Applications Symposium (RTAS), pages 258–265, Washington DC, USA, 2000.

[9] 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'05), San Francisco, USA, 2005.

[10] 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.

[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] K. Richter *Compositional Scheduling Analysis Using Standard Event Models – The SymTA/S Approach*, PhD Thesis, Technical University of Braunschweig, Germany, 2005.

[13] M. Jersak, *Compositional Performance Analysis for Complex Embedded Applications*, PhD Thesis, Technical University of Braunschweig, Germany, 2004.

[14] 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.*