

Timing Evaluation in E/E Architecture Design at BMW

Dan Gunnarsson, Matthias Traub, and Christian Pigorsch
BMW AG

ABSTRACT

Timing evaluation methods help to design a robust and extendible E/E architecture (electric/electronic). BMW has introduced the systematic application of such methods in the E/E design process within the last three years. Meanwhile, most of the architectural changes are now verified by a tool-based, automatic real-time analysis. This has increased the accuracy of the network planning and productivity of the BMW network department. In this paper, we give an overview of the actual status of timing evaluations in BMW's E/E architecture design. We discuss acceptance criteria, analysis metrics, and design rules, as far as these are related to timing. We look specifically at automation options, as these improve the productivity further. We will see that timing analysis has matured and should be mandatory for application in mass production E/E architecture development. At the same time, there is room for future improvements.

CITATION: Gunnarsson, D., Traub, M., and Pigorsch, C., "Timing Evaluation in E/E Architecture Design at BMW," *SAE Int. J. Passeng. Cars – Electron. Electr. Syst.* 7(1):2014, doi:10.4271/2014-01-0317.

INTRODUCTION

Designing E/E architectures is a challenge. Innovative functions especially in the driver assistance and infotainment do-mains (e.g. lateral collision avoidance, lane speed tracking or connected drive) are performance hungry and architectures must provide the required computation and communication capacity; at reasonable cost, weight, energy consumption, and development time. This is possible only with continuous productivity improvements in the design methods and in the overall E/E architecture development process.

Optimization and automation can help to get sustainable design decisions. Some time ago, single-bus systems were designed manually and verified by testing. Later, automation appeared with model-based testing, formal verification methods, etc. Meanwhile, model-based techniques are used also in early stages of the development process. This applies in particular to timing analysis. The timing behavior of today's heterogeneous applications, networks, and components, has become so complex that systematic (and automated) approaches for early evaluation are mandatory to avoid costly re-designs and achieve optimal and robust system designs.

State-of-the-art timing analysis tools cover the relevant real-time metrics (bus load, response times, signal latencies, gate-way delays, and etc. compare later) based on "virtual" models of the E/E architecture. The analysis forecasts the actual performance of an architecture candidate prior to its actual release. BMW has integrated such timing evaluation methods into the existing design process.

At present, the design decisions are still mostly done manually or driven by simple metrics (e.g. bus load) and architect's expertise. However, the evaluation of different design alternatives is done in detail with modern timing analysis methods. Appropriate acceptance criteria give comparable predictions when to continue and when to backtrack and optimize the alternatives. Such a timing-supported, tool-based evaluation process lifts E/E architecture design to a new level. It serves the fundamental desire for front-loading and optimization in order to minimize the risk of wrong decisions.

To extend the timing evaluation methodology BMW is currently assessing possible next steps towards further automation of the actual decision making:

- CAN id and FlexRay slot assignment, cycle time selection, signal-to-frame mapping, trigger selection
- Function distribution, software component (SW-C) mapping
- Protocol and topology selection
- Gateway configuration

Many promising methods have been proposed; many of them are self-contained and theoretically sound but rely on assumptions that conflict with the real world (e.g. legacy, heterogeneity, complexity, availability of information, design uncertainties, etc.). So, the key question is not "Which synthesis methods and metrics exist?" but "How can we embed them into the existing process?" Examples for such synthesis methods and metrics are automatic CAN identifier suggestions for new frames or Signal-to-PDU mappings in an early design phase.

During our work, it became clear that there is not the one synthesis method or metric that solves it all. Rather, it shall be possible to tune the existing synthesis and optimization methods and apply them at dedicated steps in an evolving E/E design process under real-world limitations. This leads to a semi-automated, *guided* execution of selected synthesis and optimization methods, embedded in a (still) evaluation-based design process. This in turn requires a certain flexibility and modularity in the used methods and tools. With these steps the E/E architects get the opportunity to take sustainable design decisions in every phase of the development process.

The paper is organized as follows. We review our basic E/E and network design process in the next chapter. Then, we introduce our design criteria and timing metrics, and we explain how we embedded timing analysis (tools) into our existing processes; those processes that are in use today for mass production at BMW. After that, we survey the basic ideas behind our current way of taking E/E and network design decisions, e.g. frame triggering, CAN ID select, etc. In this survey, we also elaborate on possibilities for taking such decisions automatically, by means of appropriate configuration algorithms. Finally, we summarize more advanced analysis and automation options which we will potentially consider for future process improvements. At the end, we draw our conclusions.

TIMING IN E/E ARCHITECTURE DESIGN

The present E/E architecture design and development process at BMW is iterative. The work of designing, developing and implementing an E/E architecture takes place in a number of iterations spanning from early concept phases until start of production. The actual process flow at BMW with the main steps is shown in [Figure 1](#).

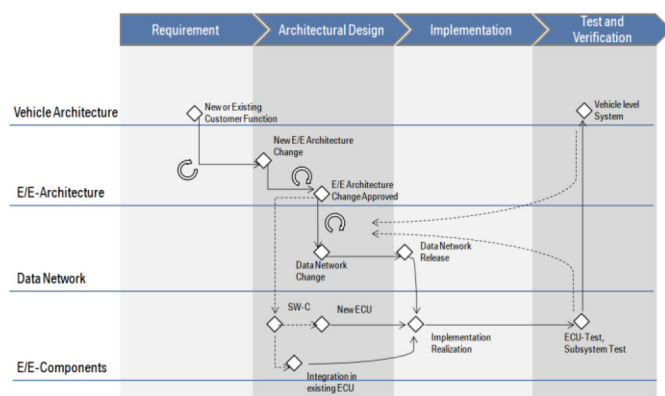


Figure 1. A typical development process at BMW

When a new function is introduced or an existing function is changed the requirements are documented in a so-called customer function. This phase is called requirement phase. In most cases the requirements are depend on a customer feature. Based on the functional architecture, the new function is then integrated into the vehicle E/E architecture, build of

ECUs and networks. This task in the architectural design phase is called “New E/E architecture change” and includes the following aspects:

- Definition of functions and new SW-Cs
- Possible re-use of existing functions
- (Re-)Allocation of functions (SW-Cs) on ECUs
- Definition or change of on-board network communication in general.

The output of this task is used as input for the next step “data network change”. If communication over the on-board networks is needed or shall be changed, we have to make some decision on the network level. Hereby the following architecture parameters need to be specified:

- Network identifier (e.g. for CAN, LIN or FlexRay messages)
- Frame triggering (e.g. periodic or event driven)
- Signal and PDU length

These parameters are chosen according to a set of rules. One of the most important decision steps is to decide if the change is feasible and can be approved from a resource and from a compatibility point of view. The result is documented in a change proposal agreed upon with all affected function owners and ECU owners.

When all changes are incorporated into the existing on-board network, the new specification is released for the implementation phase. After implementation, the validation and verification phase starts. In this phase the implementation is verified by test.

Several steps in this process (see [Figure 1](#)) have a direct or indirect influence on the system's timing. Therefore, timing aspects are considered before, during, and after certain decisions are taken.

During this process some important decisions are made regarding the implementation of the change concerning the communication. Regarding to these decisions, they are made using a set of well defined rules. Furthermore the settings of the parameters are checked with analyzing the functional requirements (e.g. timing) and using knowledge of existing functions and engineering experience. In the following the most important parameters concerning the design of the on-board communications are discussed. One of those parameters is the frame triggering. The frame triggering describes how often a frame is transmitted on a network, with a direct impact on the bus load. In the next chapter, we introduce several timing metrics that help us distinguishing a good from a bad decision with respect to timing, thereby providing guidance for the data network design process.

E/E DESIGN CRITERIA AND TIMING METRICS

In order to evaluate the proposed changes and to make optimized decisions, we have to apply standardized metrics to get comparable analysis results. Moreover, we need to define limits and guidelines of what can be accepted or not when evaluating alternative solutions. This makes it possible for an E/E architect to assess the correctness and feasibility of the proposed solution. In this paper we focus mainly on the aspects of functional timing behavior and resource consumption. on the network level.

The most important metrics to guide these decisions from our perspective are:

- Bus load.
- Worst-Case Response Time (WCRT),
- Typical-Case Response Time (TCRT)

As first and most prominent metric of all, BMW uses the **bus load** of a CAN bus. The bus (or utilization) load of a frame f is calculated quite straight-forward:

$$U_f = (L_f / S_{\text{Bus}}) / T_f$$

where L_f is the length of the frame f in bits (incl. stuff bits), S_{Bus} is the speed of the bus (in kbit/sec), and T_f is the cycle time of f . The total bus load equals the sum of loads of all frames.

At BMW, the specification of the bus is typically considered as valid if the maximum static cyclic bus load does not exceed a defined busload threshold (e.g. 50%). However, at the beginning of a vehicle development project the acceptable bus load is considerably lower in order to guarantee extensibility over the development process and life cycle of a vehicle. This very simple bus load metrics provides an initial measure how much of the available bandwidth of a CAN bus is currently used. However, it has many limitations. First, it only considers cyclic frames but has difficulties in capturing dynamic behavior of sporadically triggered frames. And second, it fails to distinguish the influences of the CAN Id (priority) on the overall communication timing. This shows us that the bus load is just a weak criterion for examining the timing behavior of frames (but it serves still as a first estimation). Because of this we have initiated a paradigm change towards more expressive timing metrics where the timing behavior of single frames is in focus.

Consequently, BMW has introduced **the worst case response time** [1] as an additional metric beside the bus load. This metric gives detailed information about the timing of each frame in contrast to generalizing by using the bandwidth metrics like bus load. Therefore a better assessment of a network change is possible where both the modeling of an individual frame and the general instability of the total bus is can be considered. At BMW the WCRT is used for functional timing-analysis to ensure that hard dead-lines are always met. When assessing a complete vehicle (and not only single

Frames) it is beneficial to use the relative WCRT. Here we use a ratio between the WCRT of a frame and its deadline. For frames with no specified deadline the cycle time is applied instead. When this is considerably smaller than 1 the frame is considered to have a sufficient specification with regard to timing relevant network parameters. In the case of relative WCRT larger than 1, frames could be dropped or they can be overwritten in the buffer of the sending ECU. To add some safety margin (or for later extensibility), we might want to use lower constraints, e.g. 70%, which means that every frame must have a response time less than 70% of its deadline or cycle. [Figure 2](#) illustrate a comparison of two network configurations A and B. Both have the same calculated bus load of 48%, and hence the bus load metrics would rate both configurations equally good. However, the response time profiles show that the frames in configuration A have far better timing properties than those in configuration B.

This metric for the WCRT gives also a prediction of the worst-case jitter (esp. for low priority frames). Thus, it can be used as a very good criterion for stability of the communication on the bus.

A third established metric is the **typical-case response time (TCRT)**. The TCRT provides numbers about the average timing behavior in addition to the sometimes conservative WCRT. This can be used to perform statistic analysis of the frequency of a certain response time occurring. Very often, TCRT and WCRT are applied both, especially when frames have hard and soft deadlines, for instance in the infotainment domain. For such use-cases a violations of the deadline with a certain frequency can often be accepted as a compromise when trying to find good and cost effective solutions.

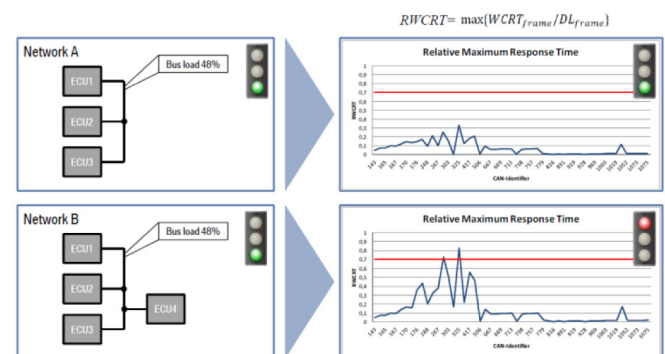


Figure 2. Comparison of two buses (Busload vs. WCRT).

Because of their great expressiveness (over the simple bus load) we are meanwhile using the mentioned metrics in the early phases of the process described in [Figure 1](#), in particular we determine the WCRT and TCRT of all frames every time a “data network change” needs to be evaluated or a “data network release” must be decided. In other words, we have enhanced our network development by systematic timing evaluations at key steps in the process (see [figure 3](#)). Of

course, we do not / cannot calculate the response times manually. Rather, we use the tool SymTA/S from Symtvision [2] which does the job for us.

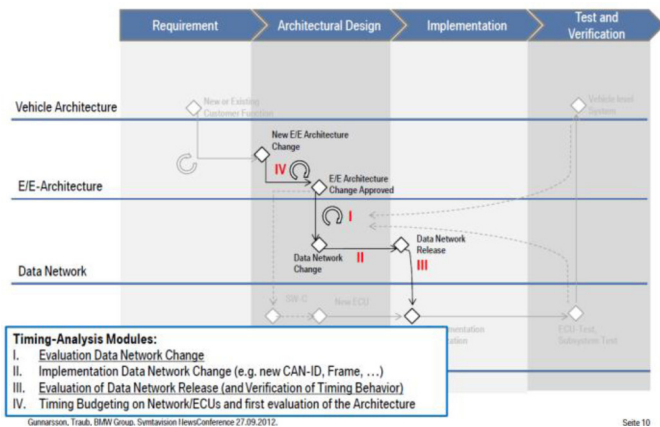


Figure 3. Timing evaluations at key steps in the process

TIMING-EFFECTIVE DESIGN RULES

In the proceeding chapter, we summarized key metrics and design criteria that help us assessing a given network configuration, if it is good or bad (from timing point of view). This chapter discusses appropriate design rules that lead us to good designs.

When an E/E architecture change is being evaluated there are several important decisions that need to be made. Here are the decisions discussed in this paper including in which stage of the above described E/E architecture process they have to be made:

- Frame Triggering Selection ("E/E Architecture Change Approved")
- CAN Identifier Selection ("Data Network Change")
- CAN Offset Selection ("Data Network Change")
- FlexRay Slot Selection ("Data Network Change")

For each of these there are different decision criteria, limitations and other information that need to be considered. These are described in the following sections.

Frame Triggering Selection

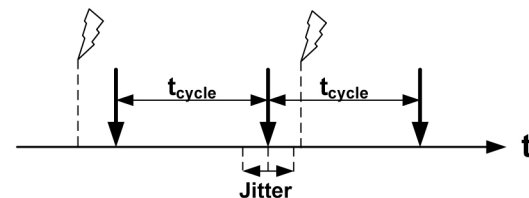
Frame triggering specifies how often a frame is transmitted over a network. There are several kinds of frame triggering such as cyclic or sporadic. To specify these triggering we need the following parameters:

- Cycle time
- Jitter including drift
- Debounce time

Cycle time specifies how often a frame shall be transmitted on the network in a periodic manner. It is specified in time units (e.g. milliseconds) between two consecutive transmissions. The

triggering is in most cases independent of functional activation, see Figure 4. To ensure a robust system design also a jitter is specified. Debounce time is the minimum time between two consecutive transmissions of an event triggered frame. This is a mechanism to protect the bus from being overloaded by multiple sporadic activations within a short time e.g. from a broken switch or babbling idiot. Both these parameters have direct influence on the amount of transmissions and the resulting load and response times on a network.

Periodic triggering



Sporadic triggering

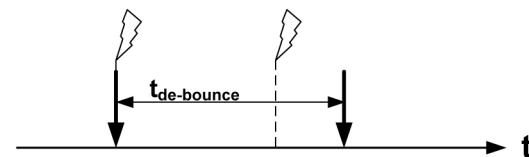


Figure 4. Periodic and sporadic frame triggering

Jitter specified how much the cycle time can be expected to maximal vary. This is useful in order to perform functional design for robust functions. Due to the small deviation of the ECU's clock from the specified cycle time this drift shall be included in the jitter. Meanwhile there exist well defined so called event models for describing the occurrence patterns frames [8], which have been standardized in the AUTOSAR Timing Extension [15].

Formerly, the frame triggering times for a change was requested by the functional architect. Due to the fact that the request was created from a functional perspective the cycle time was specified with a functional optimum as main goal. In many cases this leads to that many change requests shorter cycle time or smaller debounce time than actually needed to maximize functional performance and minimize risk of timing problems.

Thus, this approach is conflicting with one of the most important goals of an E/E network architect: using the available network resources optimally (functional, stable and cost-optimal). This means that the cycle time selection is now being performed systematically based on the functional end-to-end latency times and other known constraints as a joint effort of the E/E network architect and functional architect.

Only when a requirement can be proved to be valid and feasible a change request with certain timing parameters will be approved. Currently this task is done with simple means of rough timing budgeting. This makes it possible to easily verify many of the constraints given by the functional architect.