Document Title	Requirements on Timing Extensions
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	410
Document Classification	Auxiliary

Document Version	1.0.0
Document Status	Final
Part of Release	4.0
Revision	1

Document Change History			
Date	**Version**	**Changed by**	**Description**
30.11.2009	1.0.0	AUTOSAR Administration	Initial Release

Disclaimer

Advice for users

AUTOSAR specifications may contain exemplary items (exemplary reference models, "use cases", and/or references to exemplary technical solutions, devices, processes or software).

Any such exemplary items are contained in the specifications for illustration purposes only, and they themselves are not part of the AUTOSAR Standard. Neither their presence in such specifications, nor any later documentation of AUTOSAR conformance of products actually implementing such exemplary items, imply that intellectual property rights covering such exemplary items are licensed under the same rules as applicable to the AUTOSAR Standard.

Table of Contents

References

[1] R. Henia, A. Hamann, M. Jersak, R. Racu, K. Richter, and R. Ernst. System Level Performance Analysis - The SymTA/S Approach. IEE Proceedings Computers and Digital Techniques, 152(2): 148 to 166, March 2005.

[2] T. Pop, P. Eles, and Z. Peng. Holistic Scheduling and Analysis of Mixed Time/Event-Triggered Distributed Embedded Systems. In Proc. of the International Symposium on Hardware/Software Codesign (CODES), p. 187 to 192, New York, NY, USA, 2002.

[3] M. G. Harbour, J. J. Gutierrez Garcia, J. C. Palencia Gutierrez, and J. M. Drake Moyano. MAST: Modeling and Analysis Suite for Real-Time Applications. In Proc. Euromicro Conference on Real-Time Systems (ECRTS), p. 125, Washington, DC, USA, 2001.

[4] L. Thiele, S. Chakraborty, and M. Naedele. Real-Time Calculus for Scheduling Hard Real-Time Systems. In Proc. International Symposium on Circuits and Systems, p. 101 to 104, Geneva, Switzerland, March 2000.

1 Scope of this document

This document collects the requirements on the Timing Model and its incorporation into the AUTOSAR templates.

The main goal of the Timing Model is the extension of the AUTOSAR templates with timing information to enable the analysis and validation of a system's timing behavior.

...

The requirements collected in this document will be satisfied by the Timing Model specification (add reference to specification document). This document implements most of the requirements stated here.

2 Conventions used

Each requirement is defined as a table. The structure of the tables is as follows:

Initiator:	Initiator (e.g. WP General Methodology and Configuration)
Date:	Date of last change
Requirement:	Short description (same as above)
Description:	Detailed description
Rationale:	Why is this requirement important, what its omission could cause?
Use Case:	A scenario that makes the requirement necessary or useful
Dependencies:	References to other requirements which this requirement depends on
Conflicts:	References to other requirements which this requirement is in conflict with
Supporting Material:	References to other documents, models etc.
Comment:	Comments

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119. Note that the requirement level of the document in which they are used modifies the force of these words.

- MUST: This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.
- MUST NOT: This phrase, or the phrase "SHALL NOT", means that the definition is an absolute prohibition of the specification.
- SHOULD: This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
- SHOULD NOT: This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
- MAY: This word, or the adjective "OPTIONAL", means that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item.

An implementation, which does not include a particular option, MUST be prepared to interoperate with another implementation, which does include the option, though perhaps with reduced functionality. In the same vein an implementation, which does include a particular option, MUST be prepared to interoperate with another implementation, which does not include the option (except, of course, for the feature the option provides.)

3 Requirements

This chapter describes all requirements driving the work of the AUTOSAR Timing subgroup to define the specification of the timing model for AUTOSAR Release 4.0.

3.1 [RSTM001] Timing properties

Initiator:	WP General Methodology and Configuration, Timing Subgroup
Date:	2008-06-05
Requirement:	Timing properties
Description:	The AUTOSAR templates shall provide the means to describe the timing properties of a system's dynamics, which are determined by the consumption of computation, communication, and other hardware resources.
Rationale:	The description of timing properties in the AUTOSAR templates is an essential prerequisite for the analysis and validation of a system's timing behavior or its prediction early in the process.
Use Case:	Analysis and validation of timing behavior, early prediction of modification impacts, support for hardware dimensioning, system configuration optimization
Dependencies:	None identified.
Conflicts:	None identified.
Supporting Material:	–
Comment:	–

3.2 [RSTM002] Timing constraints

Initiator:	WP General Methodology and Configuration, Timing Subgroup
Date:	2008-06-05
Requirement:	Timing constraints
Description:	The AUTOSAR templates shall provide the means to describe timing constraints, such as software and hardware latency, input/output delay, synchronization, and runnable execution order constraints with clearly defined semantics. Also, the scope and the boundaries of timing constraints shall be explicitly described.
Rationale:	The description of timing constraints in the AUTOSAR templates is an essential prerequisite to formally capture expectations and limitations on a system's timing behavior which guide the system generation process and can be used to validate a given system configuration.
Use Case:	Analysis and validation of timing behavior, support for hardware dimensioning, system configuration optimization
Dependencies:	[RSTM004]
Conflicts:	None identified.
Supporting Material:	–
Comment:	–

3.3 [RSTM003] Optionality of timing constraints

Initiator:	WP General Methodology and Configuration, Timing Subgroup
Date:	2008-06-05
Requirement:	Optionality of timing constraints
Description:	The usage of timing constraints in the AUTOSAR templates shall be optional.
Rationale:	Usually timing constraints are only specified for a limited number of e.g. safety-related sub-systems, but not for the complete system.
Use Case:	Analysis and validation of timing behavior
Dependencies:	None identified.
Conflicts:	None identified.
Supporting Material:	–
Comment:	–

3.4 [RSTM004] Event chains

Initiator:	WP General Methodology and Configuration, Timing Subgroup
Date:	2008-06-05
Requirement:	Event chains
Description:	The AUTOSAR templates shall provide the means to describe timing specific event chains. An event chain is used as the subject to attach a timing constraint.
Rationale:	Event chains are an essential prerequisite to define the scope and semantics of timing constraints.
Use Case:	Analysis and validation of timing behavior
Dependencies:	None identified.
Conflicts:	None identified.
Supporting Material:	–
Comment:	An event chain (timing chain) describes the temporal correlation between two observable events, referred to as stimulus and response, that have a functional dependency.

3.5 [RSTM005] Structure of event chains

Initiator:	WP General Methodology and Configuration, Timing Subgroup
Date:	2008-06-05
Requirement:	Structure of event chains
Description:	It shall be possible to organize event chains in hierarchies. That is, event chains can be built up from arbitrary event sub-chains. Leaves of the hierarchy are atomic event chains. Atomic event chains are defined in the sense that stimulus and response are clearly defined by the interaction semantics.
Rationale:	A hierarchical event chain structure supports the scalability and evolvability of timing constraints.
Use Case:	Analysis and validation of timing behavior
Dependencies:	[RSTM004]
Conflicts:	None identified.
Supporting Material:	–
Comment:	–

3.6 [RSTM006] Triggering behavior of event chains

Initiator:	WP General Methodology and Configuration, Timing Subgroup
Date:	2008-06-05
Requirement:	Triggering behavior of event chains
Description:	The AUTOSAR templates shall provide the means to describe the triggering behavior (e.g. periodic, sporadic, and arbitrary) of event chains.
Rationale:	The analysis and validation of an event chain's timing constraints requires to make assumptions about the occurrence characteristics of the according stimulus and response events.
Use Case:	Analysis and validation of timing behavior
Dependencies:	[RSTM004]
Conflicts:	None identified.
Supporting Material:	–
Comment:	–

3.7 [RSTM007] Synchronization of event chains

Initiator:	WP General Methodology and Configuration, Timing Subgroup
Date:	2008-06-05
Requirement:	Synchronization of event chains
Description:	The AUTOSAR templates shall provide the means to describe timing constraints for the synchronization of multiple event chains with possibly independent stimulus and response events.
Rationale:	Synchronization is a key issue when redundant communication is considered.
Use Case:	Analysis and validation of timing behavior
Dependencies:	[RSTM002],[RSTM004]
Conflicts:	None identified.
Supporting Material:	–
Comment:	–

3.8 [RSTM008] Multiple asynchronous time bases

Initiator:	WP General Methodology and Configuration, Timing Subgroup
Date:	2008-06-05
Requirement:	Multiple asynchronous time bases
Description:	The AUTOSAR templates shall provide the means to describe multiple asynchronous clocks/time bases and their interrelation.
Rationale:	In networked systems it is reasonable to describe synchronous events even for multiple asynchronous time bases.
Use Case:	Analysis and validation of timing behavior
Dependencies:	None identified.
Conflicts:	None identified.
Supporting Material:	–
Comment:	–

3.9 [RSTM009] Loop-back signal flow in sender-receiver communication

Initiator:	WP General Methodology and Configuration, Timing Subgroup
Date:	2008-06-05
Requirement:	Loop-back signal flow in sender-receiver communication
Description:	It shall be possible to annotate conncetions among SWCs on VFB level, to indicate that a sender-receiver communication needs to be buffered.
Rationale:	When software components are connected to work together using sender-receiver communication there is a natural signal flow implied in this composition where one SW-Component produces some date which is then consumed and further processed by another software component. When such a setup also contains loop-back of signals it is no longer possible to determine which signal-flow is to be processed during one pass and which signal flow shall be buffered as the loop-back for the next execution.
Use Case:	A filter algorithm which is implemented using several software components and feeds the result of the algorithm back as an input. When this loop-back signal flow is annotated, the relationships between the other software components can be arranged in a sequence and the execution order of the involved runnable entities can be determined. Analysis and validation of timing behavior in closed loop control systems
Dependencies:	[RSTM001], [RSTM003]
Conflicts:	Non identified.
Supporting Material:	Requirements on BSW & RTE Features
Comment:	–

3.10 [RSTM010] Validity of timing properties and constraints

Initiator:	WP General Methodology and Configuration, Timing Subgroup
Date:	2008-06-05
Requirement:	Validity of timing properties and constraints
Description:	The AUTOSAR templates shall provide the means to describe the validity of timing properties and constraints, e.g. for a certain hardware or software configuration.
Rationale:	To utilize timing properties and constraints correctly, it is necessary to know the context in which they were obtained: for example a WCET is only valid for a specific implementation and target platform.
Use Case:	Analysis and validation of timing behavior
Dependencies:	[RSTM001],[RSTM002]
Conflicts:	None identified.
Supporting Material:	–
Comment:	Note, that a software and hardware context here does not necessarily imply an explicitly defined variant.

3.11 [RSTM011] Mode dependency

Initiator:	WP General Methodology and Configuration, Timing Subgroup
Date:	2008-06-05

Requirement:	Mode dependency
Description:	The AUTOSAR templates shall provide the means to describe the dependency of timing properties and constraints on operation modes defined on system and ECU level.
Rationale:	Depending on the mode a system's behavior may change, which has an impact on the system's timing characteristics.
Use Case:	Analysis and validation of timing behavior
Dependencies:	[RSTM001],[RSTM002],[RSTM010]
Conflicts:	None identified.
Supporting Material:	–
Comment:	–

3.12 [RSTM012] Sensor/actuator delay

Initiator:	WP General Methodology and Configuration, Timing Subgroup
Date:	2008-07-08
Requirement:	Sensor/actuator delay
Description:	The AUTOSAR templates shall provide the means to describe the time relation between a physical sensor acquisition (or a physical actuator change) and the availability (or provision) of the corresponding data on the port of a sensor (or actuator) software component on VFB level.
Rationale:	This information can be used to specify the time delay for the data flow between a physical sensor (or actuator) to the corresponding sensor (or actuator) software component without referring to a concrete hardware realization.
Use Case:	Analysis and validation of timing behavior
Dependencies:	[RSTM002]
Conflicts:	None identified.
Supporting Material:	–
Comment:	This is no restriction of the general latency constraint, but was introduced explicitly to support event chains on VFB level that include a sensing/actuating delay, without requiring to refer to a concrete sensor or actuator hardware realization.

4 Supported use cases

The timing information in AUTOSAR shall support the following use cases.

The functional use cases depicted within the following sections are derived by practical applications implemented and experienced within several pre-series projects. The use cases are gathered from chassis applications. They are derived from functional implementations of vehicle functions. Therefore, the following descriptions do not depict specific applications but explain common characteristics of chassis functions utilizing timing-relevant problems. These include

- timing constraints mainly driven by closed loop control characteristics
- transmission of data in equidistant time slices, forced by FlexRay bus
- calculation of application data synchronous to bus schedule

4.1 End-to-end timing

One typical use case for the information given by the timing model of AUTOSAR is timing analysis. Timing analysis is a rather general term that can be split up into two sub-activities needed to be done to obtain an overall end-to-end timing analysis. It can be distinguished between the local timing analysis of a single resource (an ECU or bus) and the global timing analysis of several interconnected resources (ECUs and busses). Timing analysis results can be used for validation by comparing analysis results with given timing constraints.

4.1.1 Local Timing Analysis (Scheduling Analysis)

An engineer might want to analyze the local timing behavior of a single resource with no respect to global dependencies. Thus, the local timing analysis addresses isolated scheduling questions regarding either a single bus or an ECU (a processor on that ECU). For example, in an early design phase this can help to get an impression of the resource utilization. Furthermore, local timing analysis can be also used for optimization purposes.

Local timing analysis is a basis for end-to-end analysis. This is addressed in the following sections.

4.1.2 Analysis of end-to-end timing in open loop control systems

Typical open loop control systems contain at least a sensor, a controller, and an actuator component. For analysis of such a control system the end-to-end timing is needed. Analysis of end-to-end timing includes:

- Identification of different event chains and alternative event chain segments.
- Analysis of end-to-end delay
- Scrutinize the impact of different execution orders on timing properties, namely the end-to-end delay.
- Determine the degrees of freedom in the event chain and/or the execution order.
- Select the most reasonable, i.e. most reliable or most effective, event chain.

4.1.3 Analysis of end-to-end timing in closed loop control systems

In comparison to the analysis of end-to-end timing in open loop control systems the closed loop control systems contain one or more feedback loops. The analysis requires the identification and description of these feedback loops and how to deal with them in terms of timing. Furthermore, the impact of the timing properties shall be analyzable.

4.1.4 Validation of end-to-end timing

Based on the results obtained during the analysis of end-to-end timing it shall be possible to validate whether the acual/given timing behaviour of the system satisfies its constraints. Concrete examples include validation of response times, buffer sizes or throughput. In all cases results of the analysis methods (e.g. [1], [2], [3], [4]) or simulation/measurement can be used to determine the system timing behaviour to be validated against the given set of timing constraints.

4.1.5 Synchronization

Synchronization in timing analysis is concerned with the time correlation of concurrent event chains within a common functional context. Two or more event chains are said to be synchronous, if the occurences of the corresponding stimulus and/or response events conincide in time with a certain predefined tolerance.

Herefore the following use cases shall be regarded.

4.1.5.1 Sensor data fusion in multi-sensor systems

A modern automobile typically features several sensors across its on-board network. These sensors can be used by many software functions. Some of them require data from different sensors simultaneously to calculate a more sophisticated correlation of sensor information.

Examples for functions that include sensor data fusion are ACC (adaptive cruise control, requiring radar and wheel data) or PDC (park distance control, requiring several sensors of one type to gain an overall environment model).

Timing analysis of this kind of functions can not only focus on isolated signal paths of each sensor for its own. The more interesting part is the synchronization of these paths involved in the function. Hence, the timing model must be capable of expressing *synchronicity constraints* for several event chains and offer the information needed for their validation.

4.1.5.2 Actuator synchronization

Additional to the before mentioned use case regarding sensor data fusion, it must be possible to synchronize actuators as well. Modern control systems contain distributed intelligent actuators, whose synchronization is crucial in order to ensure simultaneous operation. One example for an application of this use case is the synchronous door opening function. General speaking, such use cases have the common behavior that the access to the actuators is triggered by the same stimulus event. If the access shall be synchronized, then a specification of such timing constraints must be possible.

A typical example is the synchronization of the hazard warning light. First of all, event chains are specified for each indicator light, modeling the timing flow between the change of the hazard warning light switch and the state change of the indicators (e.g. blinking). The stimulus events of the several event chains are correlated, namely the change of the switch. For the response events (namely the activation of the indicator lights) it must be possible to specify a synchronization constraint in order to restrict the occurrence of the response events, i.e. to force their synchronized activation.

4.1.5.3 Bus synchronization / Gateway

There are several scenarios for Gateway synchronization. The gateway could be synchronized with one or more FlexRay bus to reduce sending or reading delays of gatways tasks. It should also be possible to synchronize several gateway activities to optimize the transmission time on the subsequent CAN.

4.2 Early prediction

The previous section has depicted general use cases, where an appropriate, timing augmented AUTOSAR meta model is an enabler for End-To-End timing analysis in different domains. In the following section regarding early prediction, we address now use cases where such an analysis framework can be used in order to enable *timing analysis during design phase*. Thus, an early prediction of the timing behavior can be done, resolving potential weak points in the design as soon as possible. The timing

validation is made based on estimates or partly known timing information in system design or specification phases.

4.2.1 Modification impacts due to adding a component

Component integration is a manifolded problem. First of all, the component to be integrated must provide timing data to enable analysis (and simulation, respectively) whether the component would fit into the target system, as well as to determine the impact on the target system. So the target system imposes some timing constraints on the component to be integrated. On the other hand the component to be integrated imposes some timing constraints on the target system in order to operate properly.

The timing uncertainty in this use-case is rather limited compared to use-case hardware dimensioning. Timing properties and most of the constraints are well known in the system already. Also the timing properties (and possible constraints) of the components that should be integrated are known at least in certain parts or can be provided on a rough estimate. Nevertheless, the impact on timing for intergration of the additional component could be analyzed and validated. Even if the new component is not yet implemented (so properties are based on completely guessed values) this use-case can avoid expensive software re-design or even hardware modifications (e.g. increased ECU-clock) in early design phases.

The main reason for this use case is the fact that due to a wrong timing behaviour the integration of new software to existing systems can lead to unexpected phenomenons like priority inversion, deadlocks and so on. Therefore, a simple calculation of the established (let's say, *70 percent CPU usage*) and the new (*20 percent*, for example) software and their addition is not an acceptable consideration of timing behavior.

4.2.2 Support for hardware dimensioning

Hardware resources[1] significantly influence the timing behavior of software. On the other hand hardware cost should be limited to the absolute minimum, since they dominate piece costs. Thus, for minimizing costs, it is straightforward to search in the hardware design space[2] to allow software components to provide timing properties that barely fullfil the timing constraints. The basic questions which are of importance could be the clock-rate of an ECU, the bandwidth of a bus or the access speed for memory modules and so on. A basic requirement is that the influence of hardware configurations for timing properties are known (At least as guessed values). If this holds, timing behavior could be validated for certain hardware settings and the minimal cost solution could be chosen in early design phases.

[1] e.g. computational power and bandwidth, but also access times for memory.
[2] for this use case a fixed system topology is assumed

4.2.3 Topology decisions

The main goal of fixing a specific topology is the optimization of the whole system with respect to predefined quality criteria. These may include maximum latencies, minimum bus load and so on. The mentioned decisions are made on the base of a state the system is in. To determine this state, analyzable information is needed which provides access to the system's characteristics.

With respect to timing, one meaningful optimization criterion is a minimum age of sensor data at its processing. Consider a FlexRay based communication system with a length of the communication cycle of 10ms. If a sensor ECU providing data each 40ms, there may be good reasons to use cycle multiplexing, assigning a certain slot at each 4^{th} communication cycle to the sensor ECU. However, reaching the goal of minimum data age, additional information is needed. Thus, defining the actual FlexRay slot to be assigned to the sensor ECU (and multiplexed with other data or - even - ECUs) needs exact knowledge about the points in time when the sensor data can be written to the buffers of the FR communication controller. This slot should be as close as possible at the time the sensor is availabe in the HW buffers. This may include estimated jitter values and release offsets referred to a certain reference event (like the *FR cycle start*), for example. Formal means for providing the mentioned information need to be defined within the upcoming concept.