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

Document Version	1.2.0
Document Status	Final
Part of Release	4.1
Revision	2

Document Change History			
Date	**Version**	**Changed by**	**Description**
2013-08-23	1.2.0	AUTOSAR Release Management	Removed requirement RS_TIMEX_00021 because it is an duplicate of RS_TIMEX_00009.
2013-02-06	1.1.0	AUTOSAR Administration	Added requirements RS_TIMEX_00013 to RS_TIMEX_00021 in order to reflect new features to be supported.
2009-11-30	1.0.0	AUTOSAR Administration	Initial Release

Disclaimer

This specification and the material contained in it, as released by AUTOSAR, is for the purpose of information only. AUTOSAR and the companies that have contributed to it shall not be liable for any use of the specification.

The material contained in this specification is protected by copyright and other types of Intellectual Property Rights. The commercial exploitation of the material contained in this specification requires a license to such Intellectual Property Rights.

This specification may be utilized or reproduced without any modification, in any form or by any means, for informational purposes only.
For any other purpose, no part of the specification may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The AUTOSAR specifications have been developed for automotive applications only. They have neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

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] Specification of Timing Extensions
AUTOSAR_TPS_TimingExtensions

[2] 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.

[3] 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.

[4] 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.

[5] 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 AUTOSAR Specification of Timing Extensions [1].

2 Conventions Used

The representation of requirements in AUTOSAR documents follows the table specified in [TPS_STDT_00078].

The verbal forms for the expression of obligation specified in [TPS_STDT_00053] shall be used to indicate requirements.

3 Use Case and Requirements Tracing

The following table list all use cases and links them to the related requirements.

Use Case	Description	Satisfied by
[UC_TIMEX_00001]	Local Timing Analysis (Scheduling Analysis)	[RS_TIMEX_00001] [RS_TIMEX_00002] [RS_TIMEX_00004] [RS_TIMEX_00005] [RS_TIMEX_00006] [RS_TIMEX_00007] [RS_TIMEX_00008] [RS_TIMEX_00010] [RS_TIMEX_00011] [RS_TIMEX_00012]
[UC_TIMEX_00002]	Analysis of end-to-end timing in open loop control systems	[RS_TIMEX_00001] [RS_TIMEX_00002] [RS_TIMEX_00004] [RS_TIMEX_00005] [RS_TIMEX_00006] [RS_TIMEX_00007] [RS_TIMEX_00008] [RS_TIMEX_00009] [RS_TIMEX_00010] [RS_TIMEX_00011] [RS_TIMEX_00012]
[UC_TIMEX_00003]	Analysis of end-to-end timing in closed loop control systems	[RS_TIMEX_00001] [RS_TIMEX_00002] [RS_TIMEX_00004] [RS_TIMEX_00005] [RS_TIMEX_00006] [RS_TIMEX_00007] [RS_TIMEX_00008] [RS_TIMEX_00009] [RS_TIMEX_00010] [RS_TIMEX_00011] [RS_TIMEX_00012]
[UC_TIMEX_00004]	Validation of end-to-end timing	[RS_TIMEX_00001] [RS_TIMEX_00002] [RS_TIMEX_00004] [RS_TIMEX_00005] [RS_TIMEX_00006] [RS_TIMEX_00007] [RS_TIMEX_00008] [RS_TIMEX_00009] [RS_TIMEX_00010] [RS_TIMEX_00011] [RS_TIMEX_00012]

Use Case	Description	Satisfied by
[UC_TIMEX_00005]	Sensor data fusion in multi-sensor systems	[RS_TIMEX_00001] [RS_TIMEX_00002] [RS_TIMEX_00004] [RS_TIMEX_00005] [RS_TIMEX_00006] [RS_TIMEX_00007] [RS_TIMEX_00008] [RS_TIMEX_00010] [RS_TIMEX_00011] [RS_TIMEX_00012]
[UC_TIMEX_00006]	Actuator synchronization	[RS_TIMEX_00001] [RS_TIMEX_00002] [RS_TIMEX_00004] [RS_TIMEX_00005] [RS_TIMEX_00006] [RS_TIMEX_00007] [RS_TIMEX_00008] [RS_TIMEX_00010] [RS_TIMEX_00011] [RS_TIMEX_00012]
[UC_TIMEX_00007]	Bus synchronization / Gateway	[RS_TIMEX_00001] [RS_TIMEX_00002] [RS_TIMEX_00004] [RS_TIMEX_00005] [RS_TIMEX_00006] [RS_TIMEX_00007] [RS_TIMEX_00008] [RS_TIMEX_00010] [RS_TIMEX_00011] [RS_TIMEX_00012]
[UC_TIMEX_00008]	Modification impacts due to adding a component	[RS_TIMEX_00001] [RS_TIMEX_00002] [RS_TIMEX_00004] [RS_TIMEX_00005] [RS_TIMEX_00006] [RS_TIMEX_00010] [RS_TIMEX_00011] [RS_TIMEX_00012]
[UC_TIMEX_00009]	Support for hardware dimensioning	[RS_TIMEX_00001] [RS_TIMEX_00002] [RS_TIMEX_00004] [RS_TIMEX_00005] [RS_TIMEX_00006] [RS_TIMEX_00007] [RS_TIMEX_00008] [RS_TIMEX_00010] [RS_TIMEX_00011] [RS_TIMEX_00012]

Use Case	Description	Satisfied by
[UC_TIMEX_00010]	Topology decisions	[RS_TIMEX_00001] [RS_TIMEX_00002] [RS_TIMEX_00004] [RS_TIMEX_00005] [RS_TIMEX_00006] [RS_TIMEX_00007] [RS_TIMEX_00008] [RS_TIMEX_00010] [RS_TIMEX_00011] [RS_TIMEX_00012]

4 Requirements

This chapter describes all requirements which are the basis for the AUTOSAR Specification of Timing Extensions.

4.1 Timing properties

[RS_TIMEX_00001] Timing properties ⌈

Type:	valid
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.
Supporting Material:	None

⌋(*UC_TIMEX_00001*, *UC_TIMEX_00002*, *UC_TIMEX_00003*, *UC_TIMEX_00004*, *UC_TIMEX_00005*, *UC_TIMEX_00006*, *UC_TIMEX_00007*, *UC_TIMEX_00008*, *UC_TIMEX_00009*, *UC_TIMEX_00010*)

4.2 Timing constraints

[RS_TIMEX_00002] Timing constraints ⌈

Type:	valid
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:	[RS_TIMEX_00004]
Supporting Material:	None

⌋(*UC_TIMEX_00001*, *UC_TIMEX_00002*, *UC_TIMEX_00003*, *UC_TIMEX_00004*, *UC_TIMEX_00005*, *UC_TIMEX_00006*, *UC_TIMEX_00007*, *UC_TIMEX_00008*, *UC_TIMEX_00009*, *UC_TIMEX_00010*)

4.3 Optionality of timing constraints

[RS_TIMEX_00003] Optionality of timing constraints ⌈

Type:	valid
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.
Supporting Material:	None

⌋

4.4 Event chains

[RS_TIMEX_00004] Event chains ⌈

Type:	valid
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. It describes the temporal correlation between two observable events, referred to as stimulus and response, that have a functional dependency.
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.
Supporting Material:	None

⌋(*UC_TIMEX_00001*, *UC_TIMEX_00002*, *UC_TIMEX_00003*, *UC_TIMEX_00004*, *UC_TIMEX_00005*, *UC_TIMEX_00006*, *UC_TIMEX_00007*, *UC_TIMEX_00008*, *UC_TIMEX_00009*, *UC_TIMEX_00010*)

4.5 Structure of event chains

[RS_TIMEX_00005] Structure of event chains ⌈

Type:	valid
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:	[RS_TIMEX_00004]
Supporting Material:	None

⌋(*UC_TIMEX_00001*, *UC_TIMEX_00002*, *UC_TIMEX_00003*, *UC_TIMEX_00004*, *UC_TIMEX_00005*, *UC_TIMEX_00006*, *UC_TIMEX_00007*, *UC_TIMEX_00008*, *UC_TIMEX_00009*, *UC_TIMEX_00010*)

4.6 Triggering behavior of event chains

[RS_TIMEX_00006] Triggering behavior of event chains ⌈

Type:	valid
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:	[RS_TIMEX_00004]
Supporting Material:	None

⌋(*UC_TIMEX_00001*, *UC_TIMEX_00002*, *UC_TIMEX_00003*, *UC_TIMEX_00004*, *UC_TIMEX_00005*, *UC_TIMEX_00006*, *UC_TIMEX_00007*, *UC_TIMEX_00008*, *UC_TIMEX_00009*, *UC_TIMEX_00010*)

4.7 Synchronization of event chains

[RS_TIMEX_00007] Synchronization of event chains ⌈

Type:	valid
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:	[RS_TIMEX_00002],[RS_TIMEX_00004]
Supporting Material:	None

⌋(*UC_TIMEX_00001*, *UC_TIMEX_00002*, *UC_TIMEX_00003*, *UC_TIMEX_00004*, *UC_TIMEX_00005*, *UC_TIMEX_00006*, *UC_TIMEX_00007*, *UC_TIMEX_00009*, *UC_TIMEX_00010*)

4.8 Multiple asynchronous time bases

[RS_TIMEX_00008] Multiple asynchronous time bases ⌈

Type:	valid
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.
Supporting Material:	None

⌋(*UC_TIMEX_00001*, *UC_TIMEX_00002*, *UC_TIMEX_00003*, *UC_TIMEX_00004*, *UC_TIMEX_00005*, *UC_TIMEX_00006*, *UC_TIMEX_00007*, *UC_TIMEX_00009*, *UC_TIMEX_00010*)

4.9 Loop-back signal flow in sender-receiver communication

[RS_TIMEX_00009] Loop-back signal flow in sender-receiver communication ⌈

Type:	valid
Description:	It shall be possible to annotate connections 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 data 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:	[RS_TIMEX_00001], [RS_TIMEX_00003]
Supporting Material:	Requirements on BSW & RTE Features

⌋(*UC_TIMEX_00002*, *UC_TIMEX_00003*, *UC_TIMEX_00004*)

4.10 Validity of timing properties and constraints

[RS_TIMEX_00010] Validity of timing properties and constraints ⌈

Type:	valid
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:	[RS_TIMEX_00001],[RS_TIMEX_00002]
Supporting Material:	None

⌋(*UC_TIMEX_00001*, *UC_TIMEX_00002*, *UC_TIMEX_00003*, *UC_TIMEX_00004*, *UC_TIMEX_00005*, *UC_TIMEX_00006*, *UC_TIMEX_00007*, *UC_TIMEX_00008*, *UC_TIMEX_00009*, *UC_TIMEX_00010*)

4.11 Mode dependency

[RS_TIMEX_00011] Mode dependency ⌈

Type:	valid
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:	[RS_TIMEX_00001],[RS_TIMEX_00002],[RS_TIMEX_00010]
Supporting Material:	None

⌋(*UC_TIMEX_00001*, *UC_TIMEX_00002*, *UC_TIMEX_00003*, *UC_TIMEX_00004*, *UC_TIMEX_00005*, *UC_TIMEX_00006*, *UC_TIMEX_00007*, *UC_TIMEX_00008*, *UC_TIMEX_00009*, *UC_TIMEX_00010*)

4.12 Sensor/actuator delay

[RS_TIMEX_00012] Sensor/actuator delay ⌈

Type:	valid
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:	[RS_TIMEX_00002]
Supporting Material:	None

⌋(*UC_TIMEX_00001*, *UC_TIMEX_00002*, *UC_TIMEX_00003*, *UC_TIMEX_00004*, *UC_TIMEX_00005*, *UC_TIMEX_00006*, *UC_TIMEX_00007*, *UC_TIMEX_00008*, *UC_TIMEX_00009*, *UC_TIMEX_00010*)

4.13 Specification of timing resources

[RS_TIMEX_00013] Specification of timing resources for software-component description ⌈

Type:	valid
Description:	One of the main criteria for mapping of SW-C onto ECUs is the resource requirement of the complete SW system assigned to a single ECU. In order to estimate the total amount of resources required for a software system, it is necessary to have the information for each of the assigned SW-Cs. Concerning this use case, the relevant CPU specific resource information is the worst case execution time. Again, please note that the overall use case can not fully be covered within the scope of this document because the final assessment of resource consumption is only feasible in the context of an ECU configuration. On the other hand, the assessment in the scope of the ECU configuration needs preparation by specifying resource claims in the scope of a software-component. This constraint is mainly caused by the fact that the relevant basic software configuration can only be available in the scope of an ECU configuration for the first time in the workflow. It is obvious that the basic software contributes to a more or less large extend to the overall resource consumption.
Rationale:	For integration purposes it is necessary to specify the worst case execution time of an executable entity in order to ensure proper integration.
Use Case:	It shall be possible to impose executim time constraints on executable entities in order to ensure temporal resource consumption.
Dependencies:	[FDUC 3.2.8]
Supporting Material:	None

⌋

4.14 Runnable Entities

[RS_TIMEX_00014] Sequence of execution of runnable entities ⌈

Type:	valid
Description:	The template must allow specifying: Constraint on the order of execution of different runnable entities
Rationale:	None
Use Case:	None
Dependencies:	[RS_SWCT_00090]
Supporting Material:	None

⌋

4.15 Timing Requirements of SW-Components

[RS_TIMEX_00015] Timing-requirements of SW-Components ⌈

Type:	valid
Description:	The SW-Component template must allow specifying the timing-requirements of each runnable entity of a SW-Component (e.g.): • Period • Reaction time
Rationale:	The SW-Component template must allow describing: • how often it has to be run • the time between a stimulus like e.g. the state change of a hardware or software entity and the expected reaction of the system (e.g. response, actuator activation)
Use Case:	None
Dependencies:	[RS_TIMEX_00013]
Supporting Material:	None

⌋

4.16 Some elements of the Timing Extensions shall be blueprintable

[RS_TIMEX_00016] Some elements of the Timing Extensions shall be blueprintable ⌈

Type:	valid
Description:	The Timing Extensions shall enable one to specifying timing constraints for elements, namely elements of port interfaces, part of the Application Interface specification.
Rationale:	The information contained in the Application Interface Specification shall be annotated with the tolerated age, periods, etc.
Use Case:	None
Dependencies:	[RS_TIMEX_00001]
Supporting Material:	None

⌋

4.17 Synchronization constraint on events

[RS_TIMEX_00017] Synchronization constraint on events ⌈

Type:	valid
Description:	The Timing Extension shall enable on to impose synchronization constraints on two or more timing descrition events.
Rationale:	While building up systems it is necessary to specify that the occurrence of events shall be synchronous, like in case of a turn signal indicator, anti blocking systems where information from several wheels are important.
Use Case:	None
Dependencies:	[RS_TIMEX_00001]
Supporting Material:	None

⌋

4.18 Predefined events for port interfaces at VFB level

[RS_TIMEX_00018] Predefined events for port interfaces at VFB level ⌈

Type:	valid
Description:	The Timing Extensions shall provide timing description event types dedicated to ports, like trigger port.
Rationale:	Due to the introduction of new types of ports, like the trigger port, the Timing Extensions shall provide specific types of timing description events to support such ports.

Use Case:	None
Dependencies:	[RS_TIMEX_00001]
Supporting Material:	None

⌋

4.19 AUTOSAR Methodology support

[RS_TIMEX_00019] AUTOSAR Methodology support ⌈

Type:	valid
Description:	The Timing Extensions shall provide means to support reuse and the usage of timing models already specified on the SW-C type level.
Rationale:	The Timing Extensions lack the support of supporting reuse in an efficient way. In order to accomplish this the Timing Extension shall provide means to make use of timing models from different timing views.
Use Case:	None
Dependencies:	[RS_TIMEX_00001]
Supporting Material:	None

⌋

4.20 Support for events indicating variable accesses

[RS_TIMEX_00020] Support for events indicating variable accesses ⌈

Type:	valid
Description:	The Timing Extension shall provide means to reference the point-in-time when executable entities, namely runnable entities, access variables.
Rationale:	In some cases it is necessary to reference the point-in-time when executable entities access variable data and being capable to impose timing constraints on these timing description events. For example, in the VFB timing view an age constraint is imposed on the point-in-time a variable data prototype is received. But there may be several executable entities accessing such variable data, but some of those executable entities may have different age timing constraints. In order to aavoid that such runnable entities are mapped to task with a higher activation frequency than required, it would be good to annotate the specific variable access.
Use Case:	None
Dependencies:	[RS_TIMEX_00001]
Supporting Material:	None

⌋

4.21 Age constraint on assembly connector

[RS_TIMEX_00021] Age constraint on assembly connector ⌈

Type:	removed
Description:	The Timing Extension shall provide means to annotate assembly connectors in order to resolve cycles in data flow and data dependencies respectively.
Rationale:	Typically, in complex systems it is often the case that SW-C produces data that is consumed by other SW-C which in turn produce data that is consumed by such SW-C again. In the simplest case, one SW-Cs produces data that is consumed by another one, and vice versa. In order to resolve such cyclic dependencies there must be a capability to annotate the producer-consumer relationship which is most important, such that the data required by one SW-C (A) is produced by the other SW-C (B) before the "'first"' SW-C (A) is executed.
Use Case:	None
Dependencies:	[RS_TIMEX_00001]
Supporting Material:	None

⌋

5 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

5.1 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 various 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.

5.1.1 Local Timing Analysis (Scheduling Analysis)

[UC_TIMEX_00001] 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.

⌋

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

[UC_TIMEX_00002] 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.

⌋

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

[UC_TIMEX_00003] 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.

⌋

5.1.4 Validation of end-to-end timing

[UC_TIMEX_00004] 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. [2], [3], [4], [5]) or simulation/measurement can be used to determine the system timing behaviour to be validated against the given set of timing constraints.

⌋

5.2 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.

5.2.1 Sensor data fusion in multi-sensor systems

[UC_TIMEX_00005] 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.

⌋

5.2.2 Actuator synchronization

[UC_TIMEX_00006] 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.

⌋

5.2.3 Bus synchronization / Gateway

[UC_TIMEX_00007] 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.

⌋

5.3 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.

5.3.1 Modification impacts due to adding a component

[UC_TIMEX_00008] 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.

⌋

5.3.2 Support for hardware dimensioning

[UC_TIMEX_00009] 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.

⌋

5.3.3 Topology decisions

[UC_TIMEX_00010] 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

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

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.

⌋

6 Change History

6.1 Change History for AUTOSAR R4.0.3 against R4.0.1

6.1.1 Removed SRS Items

None.

6.1.2 Changed SRS Items

None.

6.1.3 Added SRS Items

None.

6.2 Change History for AUTOSAR R4.1.1 against R4.0.3

6.2.1 Removed SRS Items

None.

6.2.2 Changed SRS Items

None.

6.2.3 Added SRS Items

Number	Heading
[RS_TIMEX_00013]	Specification of timing resources (formerly RS_SWCT_02050)
[RS_TIMEX_00014]	Sequence of execution of runnable entities (formerly RS_SWCT_03060)
[RS_TIMEX_00015]	Timing requirements for SW-C (formerly RS_SWCT_03080)
[RS_TIMEX_00013]	Specification of timing resources for software-component description
[RS_TIMEX_00014]	Sequence of execution of runnable entities
[RS_TIMEX_00015]	Timing-requirements of SW-Components
[RS_TIMEX_00016]	Some elements of the Timing Extensions shall be blueprintable
[RS_TIMEX_00017]	Synchronization constraint on events
[RS_TIMEX_00018]	Predefined events for port interfaces at VFB level
[RS_TIMEX_00019]	AUTOSAR Methodology support
[RS_TIMEX_00020]	Support for events indicating variable accesses
[RS_TIMEX_00021]	Age constraint on assembly connector

Table 6.1: Added Specification Items in 4.1.1

6.3 Change History for AUTOSAR R4.1.2 against R4.1.1

6.3.1 Removed SRS Items

Number	Heading
[RS_TIMEX_00021]	Age constraint on assembly connector. NOTE. Duplicate of [RS_TIMEX_00009]

Table 6.2: Removed Specification Items in 4.1.2

6.3.2 Changed SRS Items

None.

6.3.3 Added SRS Items

None.