

| | |
|-----------------------------------|---|
| Document Title | Specification of Timing Extensions for Classic Platform |
| Document Owner | AUTOSAR |
| Document Responsibility | AUTOSAR |
| Document Identification No | 411 |

| | |
|---------------------------------|------------------|
| Document Status | published |
| Part of AUTOSAR Standard | Classic Platform |
| Part of Standard Release | R24-11 |

| Document Change History | | | |
|-------------------------|---------|----------------------------|---|
| Date | Release | Changed by | Description |
| 2024-11-27 | R24-11 | AUTOSAR Release Management | <ul style="list-style-type: none"> Major rework of SL-LET Clarification of ordering of TimingDescriptionEventChain segments Annotation of imposition times to constraints |
| 2023-11-23 | R23-11 | AUTOSAR Release Management | <ul style="list-style-type: none"> Change primitive type of maxSlots/maxCycles in repetitiveEOC Improve identification and constraints of root in hierarchicalEOC Introduce Intra-LET paradigms Remove EOC.ignoreOrderAllowed |
| 2022-11-24 | R22-11 | AUTOSAR Release Management | <ul style="list-style-type: none"> Added SL-LET feature Rework document structure |
| 2021-11-25 | R21-11 | AUTOSAR Release Management | <ul style="list-style-type: none"> Removed some obsolete elements Corrected description of the attribute <code>ExecutionOrderConstraint.orderedElement</code> Corrected spelling errors and incorrect captions |
| 2020-11-30 | R20-11 | AUTOSAR Release Management | <ul style="list-style-type: none"> Added support for AUTOSAR Software Clusters and AUTOSAR Classic Platform Flexibility |





| | | | |
|------------|--------|----------------------------|---|
| 2019-11-28 | R19-11 | AUTOSAR Release Management | <ul style="list-style-type: none"> Replaced the verb "must" by the verb "shall" according to TPS_STDT_00053. Corrected spelling errors, terminology, and removed empty pages Changed attributes of TDEventFrameEthernet to be consistent with changes applied to ethernet communications. Corrected splitkey definition in timingResource. Revised section "Occurrence Expression Language for Timing Events" to improve understanding of the purpose of complex timing description event, as well as the timing functions. Changed Document Status from Final to published |
| 2018-10-31 | 4.4.0 | AUTOSAR Release Management | <ul style="list-style-type: none"> Added support for Logical Execution Time Added element SynchronizationPointConstraint Removed constraint constr_4535 from specification. Added element BswCompositionTiming |
| 2017-12-08 | 4.3.1 | AUTOSAR Release Management | <ul style="list-style-type: none"> Editorial changes in chapter 6 and 7. |
| 2016-11-30 | 4.3.0 | AUTOSAR Release Management | <ul style="list-style-type: none"> Added support for conditional timing Added support for timing constraints for Ethernet communications Added timing function to support mode dependency Minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation |





| | | | |
|------------|-------|----------------------------|--|
| 2015-07-31 | 4.2.2 | AUTOSAR Release Management | <ul style="list-style-type: none"> • Minor corrections and editorial changes • Added appendices C and D |
| 2014-10-31 | 4.2.1 | AUTOSAR Release Management | <ul style="list-style-type: none"> • Added the capability in Execution Order Constraint to reference RTE and BSW Events • Added description about how to specify time sets • Minor corrections / clarifications / editorial changes; For details please refer to the BWCStatement |
| 2014-03-31 | 4.1.3 | AUTOSAR Release Management | <ul style="list-style-type: none"> • Revised the entire contents of chapter "Application Notes" • Applied editorial changes to section "Repetitive Execution Order Constraint" |
| 2013-10-31 | 4.1.2 | AUTOSAR Release Management | <ul style="list-style-type: none"> • Clarified the semantics of jitter and removed ambiguities in the description of the Periodic Event Triggering Constraint. • Added AUTOSAR constraints in order to ensure specification of consistent Execution Order Constraints. • Added capability to specify logical successor relationships between runnable entities and groups of runnable entities. • Changed the prefix of timing functions from "ARTE" to "TIMEX" in order to be consistent with the AUTOSAR standard definitions. • Clarified the use of event types in the various timing views defined in the specification. |





| | | | |
|------------|-------|----------------------------|--|
| 2013-03-15 | 4.1.1 | AUTOSAR Release Management | <ul style="list-style-type: none"> • Applied editorial changes in order to improve readability and comprehensibility of the contents of the document • Added VFB event type <code>TDEventTrigger</code> and extended <code>TDEventSwcInternalBehaviorTypeEnum</code> to indicate variable access of runnable entities • Extended the capability of <code>SynchronizationTimingConstraint</code> to reference timing description events • Revised and extended the capabilities of <code>ExecutionOrderConstraint</code> to specify hierarchical and repetitive execution order constraints • Added the capability to specify blueprints of <code>VfbTimings</code> • Added capabilities to reference timing description events in existing timing models and to support reuse of timing models, as well as AUTOSAR methodology |
| 2011-12-22 | 4.0.3 | AUTOSAR Release Management | <ul style="list-style-type: none"> • Added new timing constraint types <code>AgeConstraint</code> and <code>ExecutionTimeConstraint</code> • Added occurrence expression language for <code>TimingDescriptionEvents</code> • Improved <code>TDEventModeDeclaration</code>, <code>BurstPatternEventTriggering</code> and <code>SwcTiming</code> |



△

| | | | |
|------------|-------|----------------------------------|--|
| 2011-04-15 | 4.0.2 | AUTOSAR Release Management | <ul style="list-style-type: none">• Dropped <code>InstanceRefs</code> and replaced with <code>ComponentInCompositionIn- stanceRef</code>• Restricted the semantics of <code>ExecutionOrderConstraint</code> and <code>OffsetTimingConstraint</code>• Parameterize the observable event 'FlexRayClusterCycleStart' by defining the cycle repetition |
| 2009-12-18 | 4.0.1 | AUTOSAR Release Management | <ul style="list-style-type: none">• Initial Release |

Disclaimer

This work (specification and/or software implementation) 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 work.

The material contained in this work is protected by copyright and other types of intellectual property rights. The commercial exploitation of the material contained in this work requires a license to such intellectual property rights.

This work 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 work may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The work has been developed for automotive applications only. It has neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Contents

| | | |
|-------------|--|----|
| 1 | Introduction | 14 |
| 1.1 | Overview | 14 |
| 1.2 | Approach | 14 |
| 1.3 | Scope | 14 |
| 1.4 | Document Conventions | 15 |
| 2 | Modeling | 18 |
| 2.1 | TimingExtensions (Timing Views) | 19 |
| 2.1.1 | VfbTiming | 20 |
| 2.1.2 | SwcTiming | 22 |
| 2.1.3 | SystemTiming | 23 |
| 2.1.4 | BswModuleTiming and BswCompositionTiming | 25 |
| 2.1.4.1 | BswModuleTiming | 25 |
| 2.1.4.2 | BswCompositionTiming | 27 |
| 2.1.5 | EcuTiming | 28 |
| 2.2 | Timing Conditions | 29 |
| 2.3 | TimingDescription | 38 |
| 2.3.1 | TimingDescriptionEventChain | 39 |
| 2.3.1.1 | Segments | 42 |
| 2.3.1.2 | Approach | 43 |
| 2.3.1.2.1 | Decomposition | 43 |
| 2.3.1.2.2 | Composition | 44 |
| 2.3.1.3 | Patterns | 45 |
| 2.3.1.3.1 | Sequence | 45 |
| 2.3.1.3.2 | Fork | 46 |
| 2.3.1.3.3 | Join | 47 |
| 2.3.1.3.4 | Alternative | 47 |
| 2.3.1.3.5 | Cycle | 48 |
| 2.3.2 | TimingDescriptionEvent | 49 |
| 2.3.2.1 | TDEventVfb | 51 |
| 2.3.2.1.1 | TDEventVfbReference | 52 |
| 2.3.2.1.2 | TDEventVfbPort | 53 |
| 2.3.2.1.2.1 | TDEventVariableDataPrototype | 54 |
| 2.3.2.1.2.2 | TDEventOperation | 56 |
| 2.3.2.1.2.3 | TDEventModeDeclaration | 58 |
| 2.3.2.1.2.4 | TDEventTrigger | 60 |
| 2.3.2.1.2.5 | Blueprinting TDEventVfbPort | 62 |
| 2.3.2.2 | TDEventSwc | 63 |
| 2.3.2.2.1 | TDEventSwcInternalBehavior | 64 |
| 2.3.2.3 | TDEventCom | 68 |
| 2.3.2.3.1 | TDEventISignal | 70 |
| 2.3.2.3.2 | TDEventIPdu | 71 |
| 2.3.2.3.3 | TDEventFrame | 73 |
| 2.3.2.3.4 | TDEventComEthernet | 75 |

| | | |
|-------------|---|-----|
| 2.3.2.3.5 | TDEventCycleStart | 77 |
| 2.3.2.3.5.1 | TDEventFrClusterCycleStart | 78 |
| 2.3.2.3.5.2 | TDEventTTCanCycleStart | 78 |
| 2.3.2.4 | TDEventBswInternalBehavior | 79 |
| 2.3.2.5 | TDEventBsw | 81 |
| 2.3.2.5.1 | TDEventBswModule | 82 |
| 2.3.2.5.2 | TDEventBswModeDeclaration | 83 |
| 2.3.2.6 | TDEventComplex | 85 |
| 2.3.2.7 | Occurrence Expression Language for Timing Events | 86 |
| 2.3.2.7.1 | Specifying an Occurrence Expression | 87 |
| 2.3.2.8 | Occurrence Expression Language Syntax | 94 |
| 2.3.2.8.1 | Interpreting an Occurrence Expression | 94 |
| 2.3.2.8.1.1 | Interpreting a Content Filter | 95 |
| 2.3.2.8.1.2 | Interpreting a Complex Event | 96 |
| 2.3.2.9 | Time Base Referencing for Timing Description Events | 98 |
| 2.4 | TimingConstraint | 98 |
| 2.4.1 | TimingConstraints on TimingDescriptionEvents | 98 |
| 2.4.1.1 | SynchronizationTimingConstraint | 100 |
| 2.4.1.2 | EventTriggeringConstraint | 101 |
| 2.4.1.2.1 | PeriodicEventTriggering | 102 |
| 2.4.1.2.2 | SporadicEventTriggering | 106 |
| 2.4.1.2.3 | ConcretePatternEventTriggering | 108 |
| 2.4.1.2.4 | BurstPatternEventTriggering | 111 |
| 2.4.1.2.5 | ArbitraryEventTriggering | 113 |
| 2.4.1.3 | OffsetTimingConstraint | 116 |
| 2.4.1.4 | AgeConstraint | 117 |
| 2.4.2 | TimingConstraints on TimingDescriptionEventChains | 118 |
| 2.4.2.1 | SynchronizationTimingConstraint | 119 |
| 2.4.2.2 | LatencyTimingConstraint | 123 |
| 2.4.3 | TimingConstraints on ExecutableEntities | 126 |
| 2.4.3.1 | ExecutionOrderConstraint | 128 |
| 2.4.3.1.1 | Ordinary Execution Order Constraint | 134 |
| 2.4.3.1.2 | Hierarchical Execution Order Constraint | 135 |
| 2.4.3.1.3 | Repetitive Execution Order Constraint | 138 |
| 2.4.3.2 | ExecutionTimeConstraint | 141 |
| 2.4.3.3 | SynchronizationPointConstraint | 143 |
| 3 | Application | 145 |
| 3.1 | Logical Execution Time | 145 |
| 3.1.1 | LET Interval Definition | 145 |
| 3.1.2 | Relation between LET Intervals | 148 |
| 3.1.3 | Executable Entity Cluster to LET Interval mapping | 153 |
| 3.1.4 | Data flow within an LET Interval | 156 |
| 3.2 | System Level Logical Execution Time | 159 |
| 3.2.1 | SL-LET Interval Definition | 159 |
| 3.2.1.1 | Execution Order | 161 |

| | | |
|----------|--|------------|
| 3.2.2 | SL-LET Time Zones | 162 |
| 3.2.2.1 | SL-LET Zone Clocks | 163 |
| 3.2.2.2 | SL-LET Zone Clock Accuracy | 164 |
| 3.3 | Classic Platform Software Clusters | 164 |
| 3.3.1 | Specifying a Dispatcher | 166 |
| 3.3.2 | Specifying Timing Parameters of a Dispatcher | 168 |
| 3.3.3 | Mapping Executable Entities to a Dispatcher | 169 |
| 3.3.4 | Mappings between Software Clusters | 169 |
| 3.3.5 | Example | 172 |
| A | Reference Material | 175 |
| A.1 | Terms and Abbreviations | 175 |
| A.2 | Imposition Times | 176 |
| A.3 | Requirements Traceability | 176 |
| B | Test Cases | 178 |
| B.1 | Component integration | 178 |
| B.1.1 | VFB view | 179 |
| B.1.2 | ECU view | 180 |
| B.2 | Engine control | 180 |
| B.2.1 | Overview | 180 |
| B.2.2 | Timing Requirements | 181 |
| B.2.3 | Formal description of timing constraints in VFB View | 182 |
| B.2.3.1 | Requirement 1 | 182 |
| B.2.3.2 | Requirement 2 | 183 |
| B.2.3.3 | Requirement 3 | 184 |
| B.2.4 | Formal description of timing constraints in ECU View | 185 |
| B.2.4.1 | Requirement 4 | 185 |
| B.2.5 | Formal description of timing constraints in SW-C View | 186 |
| B.2.5.1 | Requirement 5 | 186 |
| B.3 | Describing and Constraining Sensor and Actuator Timing | 187 |
| B.3.1 | External Event of a Sensor accessed via S/R | 188 |
| B.3.2 | External Event of an Actuator accessed via S/R | 188 |
| B.3.3 | External Event of a Sensor accessed via C/S | 188 |
| B.3.4 | External Event of an Actuator accessed via C/S | 189 |
| B.3.5 | Considering hardware I/O latency of EventChains at VFB-level | 189 |
| B.3.5.1 | Input latency | 189 |
| B.3.5.2 | Output latency | 189 |
| B.3.6 | Constraining Input or Output Latency | 190 |
| C | Examples | 191 |
| C.1 | Variant Handling | 191 |
| C.2 | Logical Execution Time | 194 |
| C.2.1 | Logical Execution Time - Timing descriptions | 194 |
| C.2.2 | Logical Execution Time - Timing constraints - Latency Timing | 194 |
| C.2.3 | Logical Execution Time - Timing constraints - Event Triggering | 194 |

| | | |
|-------|---|-----|
| C.2.4 | Logical Execution Time - Timing constraints - Event Triggering - Gap | 195 |
| C.2.5 | Logical Execution Time - Timing constraints - Offset Timing - Without | 195 |
| C.2.6 | Logical Execution Time - Timing constraints - Offset Timing - With | 196 |
| C.2.7 | Logical Execution Time - Timing constraints - Offset Timing - Arbitrary | 196 |
| C.2.8 | Logical Execution Time - Timing constraints - Offset Timing - Arbitrary Overlap | 197 |
| C.2.9 | Logical Execution Time - Timing constraints - Execution Order | 197 |
| C.3 | Software Cluster | 197 |
| C.3.1 | Software Cluster - Timing descriptions | 197 |
| C.4 | System Level Logical Execution Time | 201 |
| C.4.1 | System Level Logical Execution Time | 201 |
| C.5 | Timing Description Event Occurrence | 204 |
| C.5.1 | Timing Description Event Occurrence - Filter Expression | 204 |
| C.5.2 | Timing Description Event Occurrence - Filter Expression in ARXML | 204 |
| C.5.3 | Timing Description Event Occurrence - Grammar | 205 |
| C.6 | Application Notes | 206 |
| C.6.1 | Application Notes - VFB Elements | 206 |
| C.6.2 | Application Notes - VFB Timing - Latency Constraints | 207 |
| C.6.3 | Application Notes - VFB Timing - Execution Order Constraints | 208 |
| C.6.4 | Application Notes - COM Frame Timing | 209 |
| C.6.5 | Application Notes - COM Frame Timing - Timing Description Event Chain | 209 |
| C.6.6 | Application Notes - COM Frame Timing - Latency Constraint | 209 |
| C.7 | Application Notes - Engine Control | 209 |
| C.7.1 | Application Notes - Engine Control - Requirement 1 | 209 |
| C.7.2 | Application Notes - Engine Control - Requirement 2 | 211 |
| C.7.3 | Application Notes - Engine Control - Requirement 3 | 212 |
| C.7.4 | Application Notes - Engine Control - Requirement 4 | 214 |
| C.7.5 | Application Notes - Engine Control - Requirement 5 | 215 |
| D | Mentioned Class Tables | 217 |
| E | Splitable Elements in the Scope of this Document | 271 |
| F | Variation Points in the Scope of this Document | 272 |
| G | Change History | 273 |
| G.1 | Change History of this document according to AUTOSAR Release R4.3.0 | 273 |
| G.1.1 | Added Specification Items in Release R4.3.0 | 273 |
| G.1.2 | Changed Specification Items in Release R4.3.0 | 273 |
| G.1.3 | Added Constraints in Release R4.3.0 | 273 |
| G.1.4 | Changed Constraints in Release R4.3.0 | 273 |

| | | |
|-------|---|-----|
| G.1.5 | Deleted Constraints in Release R4.3.0 | 273 |
| G.2 | Change History of this document according to AUTOSAR Release R4.3.1 | 274 |
| G.2.1 | Added Specification Items in Release R4.3.1 | 274 |
| G.2.2 | Changed Specification Items in Release R4.3.1 | 274 |
| G.2.3 | Added Constraints in Release R4.3.1 | 274 |
| G.2.4 | Changed Constraints in Release R4.3.1 | 274 |
| G.2.5 | Deleted Constraints in Release R4.3.1 | 274 |
| G.3 | Change History of this document according to AUTOSAR Release R4.4.0 | 274 |
| G.3.1 | Added Specification Items in Release R4.4.0 | 274 |
| G.3.2 | Changed Specification Items in Release R4.4.0 | 275 |
| G.3.3 | Added Constraints in Release R4.4.0 | 275 |
| G.3.4 | Changed Constraints in Release R4.4.0 | 275 |
| G.3.5 | Deleted Constraints in Release R4.4.0 | 275 |
| G.4 | Change History of this document according to AUTOSAR Release R19-11 | 275 |
| G.4.1 | Added Specification Items in Release 19-11 | 275 |
| G.4.2 | Changed Specification Items in Release 19-11 | 275 |
| G.4.3 | Added Constraints in Release 19-11 | 276 |
| G.4.4 | Changed Constraints in Release 19-11 | 276 |
| G.4.5 | Deleted Constraints in Release 19-11 | 276 |
| G.5 | Change History of this document according to AUTOSAR Release R20-11 | 276 |
| G.5.1 | Added Specification Items in R20-11 | 276 |
| G.5.2 | Changed Specification Items in R20-11 | 276 |
| G.5.3 | Deleted Specification Items in R20-11 | 276 |
| G.5.4 | Added Constraints in R20-11 | 276 |
| G.5.5 | Changed Constraints in R20-11 | 277 |
| G.5.6 | Deleted Constraints in R20-11 | 277 |
| G.6 | Change History of this document according to AUTOSAR Release R21-11 | 277 |
| G.6.1 | Added Specification Items in R21-11 | 277 |
| G.6.2 | Changed Specification Items in R21-11 | 279 |
| G.6.3 | Deleted Specification Items in R21-11 | 279 |
| G.6.4 | Added Constraints in R21-11 | 280 |
| G.6.5 | Changed Constraints in R21-11 | 282 |
| G.6.6 | Deleted Constraints in R21-11 | 282 |
| G.7 | Change History of this document according to AUTOSAR Release R22-11 | 282 |
| G.7.1 | Added Specification Items in R22-11 | 282 |
| G.7.2 | Changed Specification Items in R22-11 | 283 |
| G.7.3 | Deleted Specification Items in R22-11 | 283 |
| G.7.4 | Added Constraints in R22-11 | 284 |
| G.7.5 | Changed Constraints in R22-11 | 286 |
| G.7.6 | Deleted Constraints in R22-11 | 286 |
| G.8 | Change History of this document according to AUTOSAR Release R23-11 | 287 |

| | | |
|-------|--|-----|
| G.8.1 | Added Specification Items in R23-11 | 287 |
| G.8.2 | Changed Specification Items in R23-11 | 287 |
| G.8.3 | Deleted Specification Items in R23-11 | 287 |
| G.8.4 | Added Constraints in R23-11 | 288 |
| G.8.5 | Changed Constraints in R23-11 | 288 |
| G.8.6 | Deleted Constraints in R23-11 | 289 |
| G.9 | Change History of this document according to AUTOSAR Release R24-11 | 289 |
| G.9.1 | Added Specification Items in R24-11 | 289 |
| G.9.2 | Changed Specification Items in R24-11 | 289 |
| G.9.3 | Deleted Specification Items in R24-11 | 290 |
| G.9.4 | Added Constraints in R24-11 | 291 |
| G.9.5 | Changed Constraints in R24-11 | 291 |
| G.9.6 | Deleted Constraints in R24-11 | 295 |

References

- [1] Meta Model
AUTOSAR_FO_MMOD_MetaData
- [2] Methodology for Classic Platform
AUTOSAR_CP_TR_Methodology
- [3] System Template
AUTOSAR_CP_TPS_SystemTemplate
- [4] Standardization Template
AUTOSAR_FO_TPS_StandardizationTemplate
- [5] Virtual Functional Bus
AUTOSAR_CP_EXP_VFB
- [6] Software Component Template
AUTOSAR_CP_TPS_SoftwareComponentTemplate
- [7] Glossary
AUTOSAR_FO_TR_Glossary
- [8] Generic Structure Template
AUTOSAR_FO_TPS_GenericStructureTemplate
- [9] Basic Software Module Description Template
AUTOSAR_CP_TPS_BSWModuleDescriptionTemplate
- [10] System-level Logical Execution Time: Augmenting the Logical Execution Time
Paradigm for Distributed Real-time Automotive Software
- [11] Requirements on Timing Extensions
AUTOSAR_FO_RS_TimingExtensions

1 Introduction

1.1 Overview

This AUTOSAR document contains the specification of the AUTOSAR Timing Extensions (TIMEX) and describes the elements of the AUTOSAR meta-model [1] used for creating timing models for the respective AUTOSAR Platform.

It is a supplement to the formal Unified Modeling Language (UML) definition of the Timing Extensions by means of the AUTOSAR meta-model. In other words, this document in addition to the formal definition provides introductory description and rationale for the part of the AUTOSAR meta-model relevant for the creation of timing models.

1.2 Approach

For the integration of timing information into the AUTOSAR meta-model there were two viable alternatives: the extension of existing templates, or the definition of a separate timing template.

Several discussions lead to the decision to explicitly NOT defining a separate timing template. The most valuable advantage of such an approach is addressed by the idea behind the current template composition. They are highly adapted to the AUTOSAR methodology ([2]) and the several templates handle specific process steps in the methodology.

It is neither the goal to provide a proposal for a timing augmented development process, nor to define an isolated, new process step (e.g. a timing process step). Therefore, the augmentation of the existing templates instead of the creation of a new timing template reduces dependencies in the meta-model among templates.

1.3 Scope

The primary purpose of TIMEX is to support constructing embedded real-time systems that satisfy given timing requirements and to perform timing analysis/validations of those systems once they have built up.

TIMEX provides a timing model as specification basis for a contract based development process, in which the development may be carried out by different organizations in different locations and time frames.

The constraints entered in the early phase of the project (when corresponding solutions are not developed yet) shall be seen as extra-functional requirements agreed between the development partners. In such way, the timing specification supports a top-down design methodology.

However, due to the fact that a pure top-down design is not feasible in most of the cases (e.g. because of legacy code), the timing specification allows the bottom-up design methodology as well.

The resulting overall specification shall enable the analysis of a system's timing behavior and the validation of the analysis results against timing constraints. Thus, timing properties required for the analysis shall be contained in the timing augmented system model. Example: the priority of a task, the activation behavior of an interrupt, the sender timing of a PDU and frame etc.).

Such timing properties can be found all across AUTOSAR. For example [3] provides means to configure and specify the timing behavior of the communication stack. Furthermore the execution time of executable entities can be specified. In addition, the overall specification shall provide means to describe timing constraints. A timing constraint defines a restriction for the timing behavior of the system (e.g. bounding the maximum latency from sensor sampling to actuator access). Timing constraints are added to the system model using TIMEX. Constraints, together with the result of timing analysis, are considered during the validation of a system's timing behavior, when a nominal/actual value comparison is performed.

Note: TIMEX enables the analysis and validation of an AUTOSAR system's timing behavior. However, the specification of analysis and validation **results** (e.g. the maximum resource load of an ECU, etc.) is not addressed in this document.

1.4 Document Conventions

Technical terms are typeset in mono spaced font, e.g. `PortPrototype`. As a general rule, plural forms of technical terms are created by adding "s" to the singular form, e.g. `PortPrototypes`. By this means the document resembles terminology used in the AUTOSAR XML Schema.

This document contains constraints in textual form that are distinguished from the rest of the text by a unique numerical constraint ID, a headline, and the actual constraint text starting after the [character and terminated by the] character.

The purpose of these constraints is to literally constrain the interpretation of the AUTOSAR meta-model such that it is possible to detect violations of the standardized behavior implemented in an instance of the meta-model (i.e. on M1 level).

Makers of AUTOSAR tools are encouraged to add the numerical ID of a constraint that corresponds to an M1 modeling issue as part of the diagnostic message issued by the tool.

The attributes of the classes introduced in this document are listed in form of class tables. They have the form shown in the example of the top-level element AUTOSAR:

Please note that constraints are not supposed to be enforceable at any given time in an AUTOSAR workflow. During the development of a model, constraints may legitimately be violated because an incomplete model will obviously show inconsistencies.

However, at specific points in the workflow, constraints shall be enforced as a safeguard against misconfiguration.

The points in the workflow where constraints shall be enforced, sometimes also known as the "binding time" of the constraint, are different for each model category, e.g. on the classic platform, the constraints defined for software-components are typically enforced prior to the generation of the RTE while the constraints against the definition of an Ecu extract shall be applied when the Ecu configuration for the Com stack is created.

For each document, possible binding times of constraints are defined and the binding times are typically mentioned in the constraint themselves to give a proper orientation for implementers of AUTOSAR authoring tools.

Let [AUTOSAR](#) be an example of a typical class table. The first rows in the table have the following meaning:

Class: The name of the class as defined in the UML model.

Package: The UML package the class is defined in. This is only listed to help locating the class in the overall meta model.

Note: The comment the modeler gave for the class (class note). Stereotypes and UML tags of the class are also denoted here.

Base Classes: If applicable, the list of direct base classes.

The headers in the table have the following meaning:

Attribute: The name of an attribute of the class. Note that AUTOSAR does not distinguish between class attributes and owned association ends.

Type: The type of an attribute of the class.

Mul.: The assigned multiplicity of the attribute, i.e. how many instances of the given data type are associated with the attribute.

Kind: Specifies, whether the attribute is aggregated in the class (`aggr` aggregation), an UML attribute in the class (`attr` primitive attribute), or just referenced by it (`ref` reference). Instance references are also indicated (`iref` instance reference) in this field.

Note: The comment the modeler gave for the class attribute (role note). Stereotypes and UML tags of the class are also denoted here.

Please note that the chapters that start with a letter instead of a numerical value represent the appendix of the document. The purpose of the appendix is to support the explanation of certain aspects of the document and does not represent binding conventions of the standard.

The verbal forms for the expression of obligation specified in [TPS_STDT_00053] shall be used to indicate requirements, see Standardization Template, chapter Support for Traceability ([4]).

The representation of requirements in AUTOSAR documents follows the table specified in [TPS_STDT_00078], see Standardization Template, chapter Support for Traceability ([4]).

2 Modeling

The AUTOSAR Timing Extensions (TIMEX) provide some basic means to describe and specify timing information: [TimingDescriptions](#), expressed by [TimingDescriptionEvents](#) and [TimingDescriptionEventChains](#), and [TimingConstraints](#) that are imposed on these [TimingDescriptionEvent](#) and [TimingDescriptionEventChains](#). Both means, [TimingDescriptions](#) and [TimingConstraints](#), are organized in *timing views* for specific purposes.

In general, the purpose of TIMEX are two fold:

- to provide timing requirements that guide the construction of systems which eventually shall satisfy those timing requirements
- to provide sufficient timing information to analyze and validate the temporal behavior of a system

Compared to the specification of a system's functional behavior, the specification of its timing behavior requires additional information to be captured. Not only the eventual occurrence of events but also their exact timing or the concurrency of various events become important. Therefore, in the specification of timing extensions for AUTOSAR, the *event* is the basic entity. This event is used to refer to an observable behavior within a system at a certain point in time. Example: (e.g. the activation of a [RunnableEntity](#), the transmission of a frame etc.)

Having to deal with different abstraction levels and views (see chapter [2.1](#)), and in order to avoid semantic confusion with existing concepts, a new abstract type [TimingDescriptionEvent](#) (see section [2.3.2](#)) is introduced as a formal basis for the timing extensions. Depending on the model entity and the associated observable behavior, specific timing events are defined and linked to the different views.

For the analysis of a system's timing behavior usually not only single events but also the correlation of different events is of fundamental importance. To relate timing events to each other, a further concept called [TimingDescriptionEventChain](#) (see section [2.3.1](#)) is introduced. Hereby, it is important to note that for the referenced events of an event chain a functional dependency is implicitly assumed. This means that an event of a chain somehow causes subsequent chain events. An example for an end-to-end event chain with bus communication is depicted in Figure [2.5](#) in chapter [2.1](#). This event chain describes the path from software component instance "SWC1" to software component instance "SWC3".

Based on events and event chains, it is possible to express various specific timing constraints derived from the abstract type [TimingConstraint](#). These timing constraints specify the expected timing behavior. As timing constraints shall be valid independently from implementation details, they are also expressed on a abstract level by referencing the above introduced formal basis of [TimingDescriptionEvents](#) and [TimingDescriptionEventChains](#).

Thus, by means of events, event chains and timing constraints defined on top of these, a separate central timing specification can be provided, decoupling the expected timing behavior from the actually implemented behavior. This approach supports timing contracts for AUTOSAR systems in a top-down as well as bottom-up approach.

This chapter shall walk through the meta-model representation of the timing extensions in the following sub-sections.

2.1 TimingExtensions (Timing Views)

A TIMEX model starts with the meta-class [TimingExtension](#) or rather, one of the sub-classes of [TimingExtension](#) as the top-level element. This is the owning element for all other related elements. The sub-classes of [TimingExtension](#) define a set of timing views as shown in Figure 2.1 and detailed in the next sub-sections. The chosen view depends on the availability of necessary information and the stage in the workflow. The timing views are:

- **VfbTiming**: timing information related to the interaction (e.g. reading and writing of data to ports) of [SwComponentType](#)s at VFB level (2.1.1)
- **SwcTiming**: timing information related to the [SwcInternalBehavior](#) of [AtomicSwComponentType](#)s (2.1.2)
- **SystemTiming**: timing information related to a [System](#), utilizing information about topology, software deployment, and signal mapping (2.1.3)
- **BswModuleTiming/BswCompositionTiming**: respectively, timing information related to the [BswInternalBehavior](#) of a single [BswModuleDescription](#); timing information related to [BswInternalBehavior](#) of more than one implementation of a [BswModuleDescription](#) (2.1.4)
- **EcuTiming**: timing information related to the [EcucValueCollection](#) covering the entire software on an ECU with application software and configured basic software (2.1.5)

The event types are used to specify:

- reading and writing of data from and to specific SWC ports
- calling of services and receiving their responses (VFB, SW-C, System and ECU Timing)
- sending and receiving data via networks and through communication stacks (System and ECU Timing)
- activating, starting and terminating executable entities (SW-C Timing and Basic SW Module Timing)
- calling basic software services and receiving their responses (ECU Timing and Basic SW Module Timing).

[TPS_TIMEX_00009] Optionality of Timing Extensions

Upstream requirements: [RS_TIMEX_00003](#)

To maintain its optional use in the AUTOSAR methodology, TIMEX always references "outwards" towards other external (i.e. in other template specifications) model elements.]

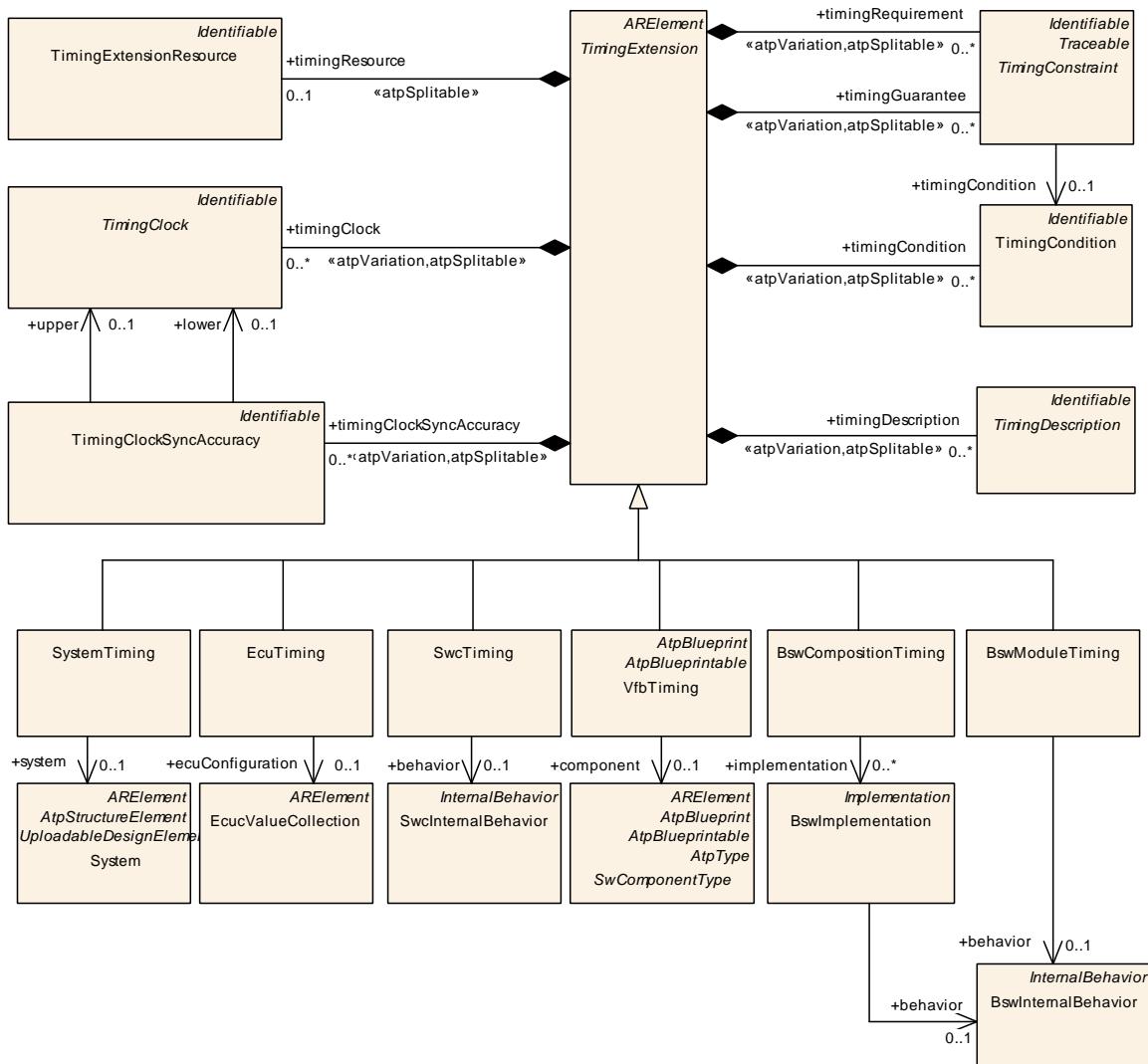


Figure 2.1: Timing Extensions top-level view

2.1.1 VfbTiming

AUTOSAR defines the *Virtual Functional Bus* [5] as a composition of [SwComponentPrototypes](#) at a logical level, regardless of their physical distribution. On this logical level a special view can be applied for timing specification. This section describes what kind of timing specification can be applied at VFB level for a system or sub-system. Typ-

ically, end-to-end timing constraints, including (physical) sensors and actuators, shall be captured in this view, allowing an early formalization of those constraints.

Neglecting the physical distribution means that the [VfbTiming](#) view does not deal with the question, in which system context the prototype of a [CompositionSwComponentType](#) shall be implemented. An additional restriction of the [VfbTiming](#) view is present due to the black box treatment of software components. The [SwcInternalBehavior](#) of [AtomicSwComponentTypes](#) is not considered. For these mentioned restrictions (irrelevance of the physical distribution, black box view), [TimingDescriptions](#) at VFB level should only refer to [SwComponentTypes](#), [PortPrototypes](#) and their connections, but not the [InternalBehavior](#).

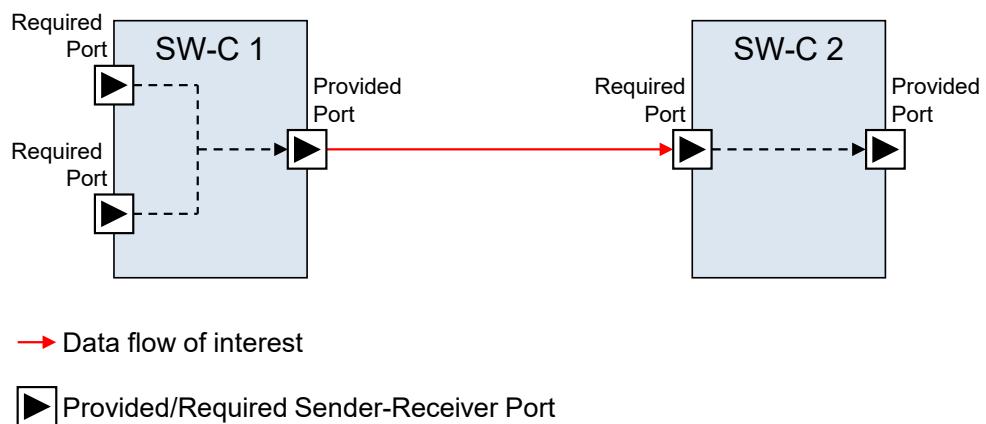
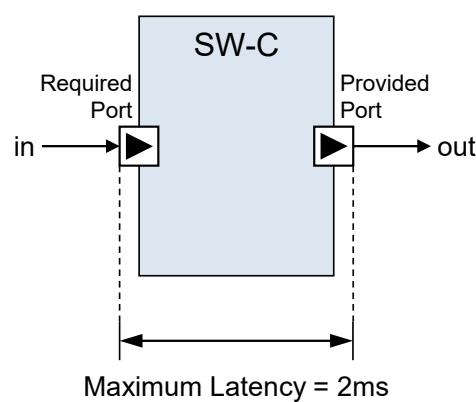


Figure 2.2: Example: Data flow in the scope of the VfbTiming view

The [VfbTiming](#) view is applicable for different system granularities. The smallest granularity is the investigation of a single [SwComponentType](#) without any contextual embedding. Here, a timing description can only refer to relations between a component's [RPortPrototypes](#) and the same component's [PPortPrototypes](#).



► Provided/Required Sender-Receiver Port

Figure 2.3: Example: Latency requirement

As an example, consider the timing constraint illustrated in Figure 2.3: "From the point in time, where the value *in* is received by the SWC, until the point in time, where the

newly calculated value *out* is sent, there shall be a maximum latency of 2 ms". This would be attached to the timing description that refers to an [AtomicSwComponentType](#) *SW-C*.

In case of a [CompositionSwComponentType](#) that itself contains other [SwComponentPrototypes](#), the timing interrelation between different components, e.g. from one component's [PPortPrototype](#) to another component's [RPortPrototype](#), could be of interest.

[TPS_TIMEX_00032] Purpose of [VfbTiming](#)

Upstream requirements: [RS_TIMEX_00001](#)

[The element [VfbTiming](#) aggregates all timing information, timing descriptions and timing constraints related to the [VFB View](#).]

| | | | | |
|----------------------|--|--------------|-------------|---|
| Class | VfbTiming | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingExtensions | | | |
| Note | A model element used to define timing descriptions and constraints at VFB level. TimingDescriptions aggregated by VfbTiming are restricted to event chains referring to events which are derived from the class TDEventVfb . Tags: <code>atp.recommendedPackage=TimingExtensions</code> | | | |
| Base | <code>ARElement, AROObject, AtpBlueprint, AtpBlueprintable, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable, TimingExtension</code> | | | |
| Aggregated by | ARPackage.element | | | |
| Attribute | Type | Mult. | Kind | Note |
| component | SwComponentType | 0..1 | ref | This defines the scope of a VfbTiming . All corresponding timing descriptions and constraints shall be defined within this scope. |

Table 2.1: [VfbTiming](#)

[constr_6848] Existence of [VfbTiming.component](#)

Imposition time: [IT_VfbTd](#)

[For each [VfbTiming](#), the reference to [SwComponentType](#) in the role *component* shall exist]

2.1.2 [SwcTiming](#)

In contrast to the [VfbTiming](#) view, a specification engineer might especially be interested in the [SwcInternalBehavior](#) of [AtomicSwComponentType](#)s that are represented as black boxes at VFB level. The [SwcInternalBehavior](#) specifies a component's behavioral decomposition into runnable entities, which are executed at runtime. Thus, in [SwcTiming](#) view, a timing description is attached to the [SwcInternalBehavior](#) of a [SwComponentType](#). It can refer to the activation, start, and termination (see section [2.3.2.2.1](#)) of the execution of a [RunnableEntity](#).

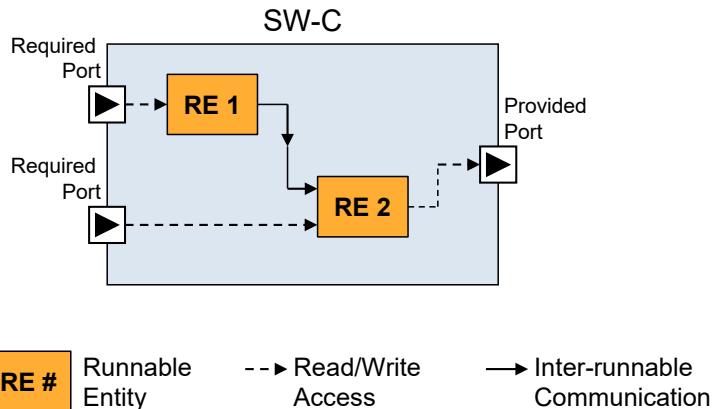


Figure 2.4: Example: Data flow in the scope of the SW-C Timing view

[TPS_TIMEX_00033] Purpose of SwcTiming

Upstream requirements: RS_TIMEX_00001

[The element `SwcTiming` aggregates all timing information, timing descriptions and timing constraints related to the Software Component View.]

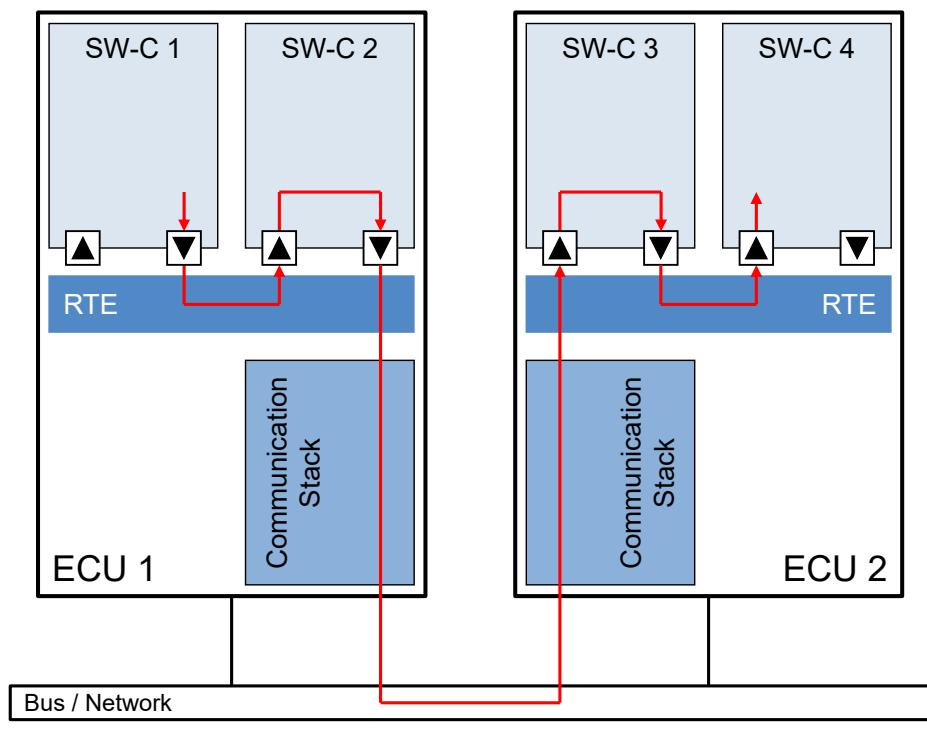
| Class | SwcTiming | | | |
|---------------|---|-------|------|---|
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingExtensions | | | |
| Note | The SwcTiming is used to describe the timing of an atomic software component. TimingDescriptions aggregated by SwcTiming are restricted to event chains referring to events which are derived from the classes TDEventVfb and TDEventSwInternalBehavior. Tags: atp.recommendedPackage=TimingExtensions | | | |
| Base | <i>ARElement, AROObject, CollectableElement, Identifiable, MultilanguageReferrable, Packageable Element, Referrable, TimingExtension</i> | | | |
| Aggregated by | <code>ARPackage.element</code> | | | |
| Attribute | Type | Mult. | Kind | Note |
| behavior | <code>SwInternalBehavior</code> | 0..1 | ref | This defines the scope of a SwcTiming. All corresponding timing descriptions and constraints shall be defined within this scope. Note! The reason for the cardinality of 0..1 is to ensure backward compatibility. |

Table 2.2: SwcTiming

2.1.3 SystemTiming

At system level a special prototype of a `CompositionSwComponentType`—the `RootSwCompositionPrototype`—is instantiated. This prototype, the chosen hardware topology and other artifacts are used as input to the task dealing with the deployment of software components onto ECUs in order to configure the system. The main configuration result is the mapping of software components to ECUs and in further steps the resulting communication matrix is created. This information is aggregated in the `System` description.

The [SystemTiming](#) view is used to provide timing information at system level. As an extension, it can be attached to a [System](#). As the [System](#) description aggregates all the information about [SwComponentTypes](#) and their corresponding [SwcInternalBehavior](#), it is possible to use the same concepts that are available in the views [VfbTiming](#) and [SwcTiming](#) also in this timing view. The difference is the specific system context that defines the validity of timing information at system level. Without knowledge of the mapping of software components to a target hardware respectively ECU, only a generic platform independent description can be provided.



→ Data flow of interest

Figure 2.5: Example: Data flow in the scope of System Timing view

In addition, a timing description in system view refers to the concrete communication of software components that only was represented as abstract connectors in [VfbTiming](#) view. Due to the software mapping, now communication is either local communication over the RTE (both software components on same ECU) or remote communication over the RTE, through the communication stack of the BSW and a communication bus. A system-specific timing description thus can refer to signals (RTE), I-PDUs (COM) and frames (communication driver and bus).

[TPS_TIMEX_00034] Purpose of [SystemTiming](#)

Upstream requirements: [RS_TIMEX_00001](#)

[The element [SystemTiming](#) aggregates all timing information, timing descriptions and timing constraints, that is related to the System View.]

| Class | SystemTiming | | | |
|---------------|---|-------|------|---|
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingExtensions | | | |
| Note | A model element used to refine timing descriptions and constraints (from a VfbTiming) at System level, utilizing information about topology, software deployment, and signal mapping described in the System Template. TimingDescriptions aggregated by SystemTiming are restricted to events which are derived from the class TDEventVfb, TDEventSwInternalBehavior and TDEventCom. Tags: atp.recommendedPackage=TimingExtensions | | | |
| Base | <i>ARElement, AROObject, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable, TimingExtension</i> | | | |
| Aggregated by | ARPackage.element | | | |
| Attribute | Type | Mult. | Kind | Note |
| system | System | 0..1 | ref | This defines the scope of a SystemTiming. All corresponding timing descriptions and constraints shall be defined within this scope. |

Table 2.3: SystemTiming

[constr_6849] Existence of [SystemTiming.system](#)

Imposition time: [IT_SysTd](#)

[For each [SystemTiming](#), the reference to [System](#) in the role [system](#) shall exist]

2.1.4 BswModuleTiming and BswCompositionTiming

The Basic Software Timing consists of two perspectives, which are described in the next sections.

2.1.4.1 BswModuleTiming

As part of the methodology, a [BswModuleDescription](#) is generated for each BSW module as part of the ECU configuration phase. For every module its internals, the [BswInternalBehavior](#), shall be defined, i.e. structuring any [BswModuleEntity](#). Similar to the timing view on the [SwcInternalBehavior](#) of an [AtomicSwComponentType](#) as described in section 2.1.2, the BSW module timing view focuses on the activation, start and end of the execution of any [BswModuleEntity](#). The timing description for each module can be attached to the [BswModuleDescription](#).

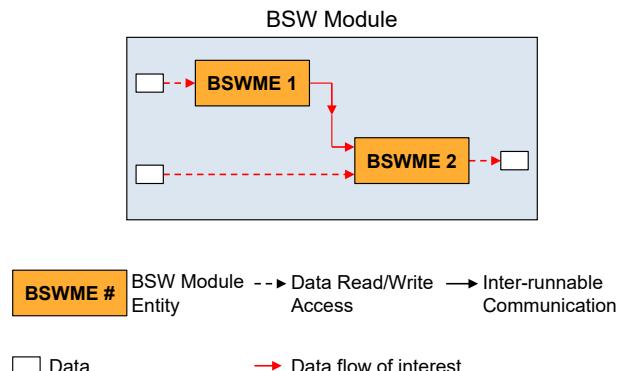


Figure 2.6: Example: Data flow in scope of BSW Module Timing view

[TPS_TIMEX_00035] Purpose of `BswModuleTiming`

Upstream requirements: RS_TIMEX_00001

〔The element `BswModuleTiming` aggregates all timing information, timing descriptions and timing constraints, that is related to the Basic Software Module View.〕

| | | | | |
|----------------------|---|--------------|-------------|--|
| Class | <code>BswModuleTiming</code> | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingExtensions | | | |
| Note | <p>A model element used to define timing descriptions and constraints for the <code>BswInternalBehavior</code> of one BSW Module. Thereby, for each <code>BswInternalBehavior</code> a separate timing can be specified.</p> <p>A constraint defined at this level holds true for all Implementations of that <code>BswInternalBehavior</code>.</p> <p>TimingDescriptions aggregated by <code>BswModuleTiming</code> are restricted to event chains referring to events which are derived from the class <code>TDEventBswInternalBehavior</code>.</p> <p>Tags: atp.recommendedPackage=TimingExtensions</p> | | | |
| Base | ARElement, AROObject, CollectableElement, <i>Identifiable</i> , MultilanguageReferrable, Packageable Element, <i>Referable</i> , <i>TimingExtension</i> | | | |
| Aggregated by | <code>ARPackage.element</code> | | | |
| Attribute | Type | Mult. | Kind | Note |
| behavior | <code>BswInternalBehavior</code> | 0..1 | ref | This defines the scope of a <code>BswModuleTiming</code> . All corresponding timing descriptions and constraints shall be defined within this scope. |

Table 2.4: `BswModuleTiming`

[constr_6850] Existence of `BswModuleTiming.behavior`

Imposition time: IT_BswTd

〔For each `BswModuleTiming`, the reference to `BswInternalBehavior` in the role `behavior` shall exist〕

2.1.4.2 BswCompositionTiming

In contrast to the element [BswModuleTiming](#), which describes the timing information of a single [BswInternalBehavior](#), the [BswCompositionTiming](#) describes the timing information for more than one implementation of a basic software module — a *composition* of basic software modules. Typically, this is the case when the timing information of a basic software stack, consisting of several basic software modules, shall be described or the timing information of a basic software module that is deployed onto different cores shall be described.

[TPS_TIMEX_00053] Purpose of [BswCompositionTiming](#)

Upstream requirements: [RS_TIMEX_00001](#)

〔The element [BswCompositionTiming](#) describes the timing information related to [BswInternalBehavior](#) of more than one implementation of a [BswModuleDescription](#).〕

| | | | | |
|----------------------|--|--------------|-------------|---|
| Class | BswCompositionTiming | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingExtensions | | | |
| Note | <p>A model element used to define timing descriptions and constraints for a set of BswImplementations representing a BSW composition. A constraint defined at this level holds true for all referenced Bsw Implementations. Note, that multiple implementations of the same basic software module could be involved.</p> <p>TimingDescriptions aggregated by BswCompositionTiming are restricted to event chains referring to events which are derived from the class TDEventBswInternalBehavior and TDEventBsw.</p> <p>Tags: atp.recommendedPackage=TimingExtensions</p> | | | |
| Base | ARElement, AROObject, CollectableElement, Identifiable , MultilanguageReferrable, Packageable Element, Referrable , TimingExtension | | | |
| Aggregated by | ARPackage.element | | | |
| Attribute | Type | Mult. | Kind | Note |
| implementation | BswImplementation | * | ref | This defines the scope of a BswCompositionTiming. All corresponding timing descriptions and constraints shall be defined within this scope. |

Table 2.5: [BswCompositionTiming](#)

[constr_6851] Existence of [BswCompositionTiming.implementation](#)

Imposition time: [IT_BswTd](#)

〔For each [BswCompositionTiming](#), the reference to [BswImplementation](#) in the role [implementation](#) shall exist at least once〕

2.1.5 EcuTiming

A result of the ECU configuration phase is the complete [EcucValueCollection](#) representing the ECU's configuration description. During ECU configuration, this artifact is filled amongst others with:

- the ECU Extract of System Configuration, where the needed part of the overall system description for the respective ECU is extracted.
- references to information about all BSW modules present on the ECU. Such BSW modules are described by [BswModuleDescriptions](#), providing for instance information about the interfaces that the modules offer or require.

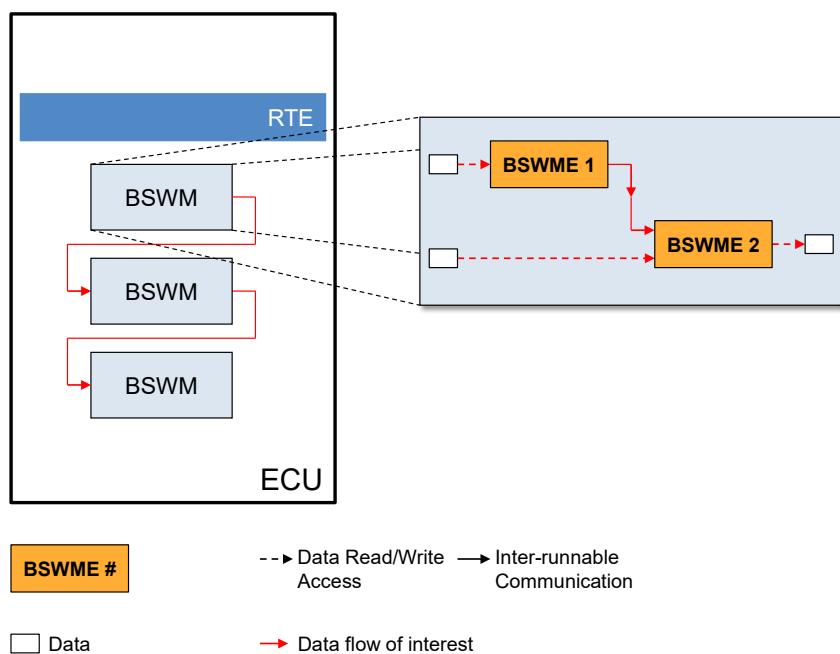


Figure 2.7: An example of data flow, whose timing behavior is in scope of ECU view

In this view, timing can reference all the ECU-relevant information: The deployed software component instances, the ECU related interactions including bus communication, Basic Software, etc. In other words, the [EcuTiming](#) has the same expressivity as the System Timing view but only focusing on one specific ECU in the system's topology. In addition, the entire BSW can be considered during timing modeling, because the complete composition and internal structure of the BSW modules are known. The internals of BSW modules and the inter-relations between BSW modules are of interest in this timing view. The information is attached to the [EcucValueCollection](#).

[TPS_TIMEX_00036] Purpose of [EcuTiming](#)

Upstream requirements: [RS_TIMEX_00001](#)

[The element [EcuTiming](#) aggregates all timing information, timing descriptions and timing constraints, that is related to the ECU View.]

| Class | EcuTiming | | | |
|-------------------|---|-------|------|---|
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingExtensions | | | |
| Note | A model element used to define timing descriptions and constraints within the scope of one ECU configuration. TimingDescriptions aggregated by EcuTiming are allowed to use all events derived from the class Timing DescriptionEvent. Tags: atp.recommendedPackage=TimingExtensions | | | |
| Base | <i>ARElement, ARObject, CollectableElement, Identifiable, MultilanguageReferrable, Packageable Element, Referrable, TimingExtension</i> | | | |
| Aggregated by | <i>ARPackage.element</i> | | | |
| Attribute | Type | Mult. | Kind | Note |
| ecu Configuration | <i>EcucValueCollection</i> | 0..1 | ref | This defines the scope of an EcuTiming. All corresponding timing descriptions and constraints shall be defined within this scope. |

Table 2.6: EcuTiming

[constr_6852] Existence of *EcuTiming.ecuConfiguration*

Imposition time: *IT_EcuTd*

[For each *EcuTiming*, the reference to *EcucValueCollection* in the role *ecu-Configuration* shall exist at least once]

2.2 Timing Conditions

In almost all cases, systems, and the application software executed in those systems, operate under various conditions, like normal condition, error condition, start-up condition, etc. During the operation of such systems the conditions may change at any time.

As a consequence timing constraints imposed on the system may vary depending on these conditions, too. TIMEX provides a means to support the description of timing constraints depending on such conditions by *Conditional Timing* as described in this section.

Almost all software management systems controlling an internal combustion engine shall maintain a constant temperature of the coolant to ensure the optimal operation under specific conditions. For example, critical thermal conditions may occur that may lead to severe damage of the engine's mechanical components. In order to prevent the engine's mechanical components from being damaged the software application shall respond faster than in the nominal case. Therefore, different timing constraints are imposed on a given event chain depending on a known condition.

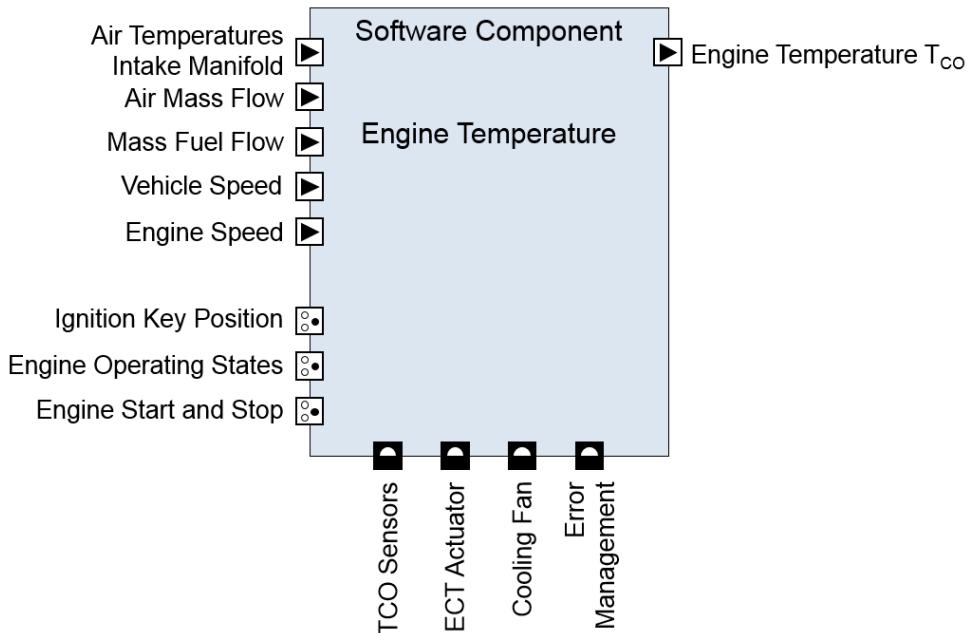


Figure 2.8: Rough sketch of an AUTOSAR software component (SW-C) controlling the coolant temperature (TCO) of an internal combustion engine. This SW-C shall consider various modes of operation as shown in the lower left corner of the components, namely the mode ports called "Ignition Key Position", "Engine Operating States", and "Engine Start and Stop".

As shown in Figure 2.8 the SWC controlling the coolant temperature requires information on the current operating conditions of the internal combustion engine. Depending on these conditions different timing constraints are imposed on an event chain requiring different reaction time constraints.

In one situation, the reaction time shall be much faster than in other situation. For example, the mode port called Engine Operating States indicates a specific condition respectively *mode* the internal combustion engine operates in. In this specific mode it may happen that the internal combustion engine faces a critical thermal condition.

In order to prevent the engine from being damaged the Electronically Controlled Thermostat (ECT) shall be actuated much faster (client port called ECT Actuator) than in other cases. This is accomplished by specifying a tighter reaction time for this condition; and under any other condition the reaction time may be more relaxed.

Besides this particular example, there are a lot more cases where the timing constraints imposed on given event chains change depending on the internal combustion engines mode.

In dependable systems an application is typically designed in a fault tolerant manner. It either operates under normal condition — normal mode — or under failure condition — failure mode. As sketched in the upper part of Figure 2.9 in such an application a number of runnable entities process the values of three sensors. In a first step the sensor values are corrected by a runnable entity and are passed to another runnable

entity of the application that utilizes a control algorithm to calculate an output. This output value is used to control an actuator or is processed by other runnable entities in the system. In the same application another runnable entity checks the plausibility of the values received from the sensors.

The purpose of this runnable entity is to determine any unexpected and implausible deviation from the expected values of each sensor. If such a deviation is detected by the runnable entity checking the plausibility of the sensor values then a runnable entity correcting the sensor values is notified about this condition, namely indicating a failure of one of the sensors. The runnable entity correcting the sensor values is capable of deriving the value of the erroneous sensor based on the values of the other sensors which monitor other physical properties.

Thus, the application is operating under different conditions — normal and failure mode — and with regard to the software implementation this results in different worst case execution times of the runnable entity correcting the sensor values.

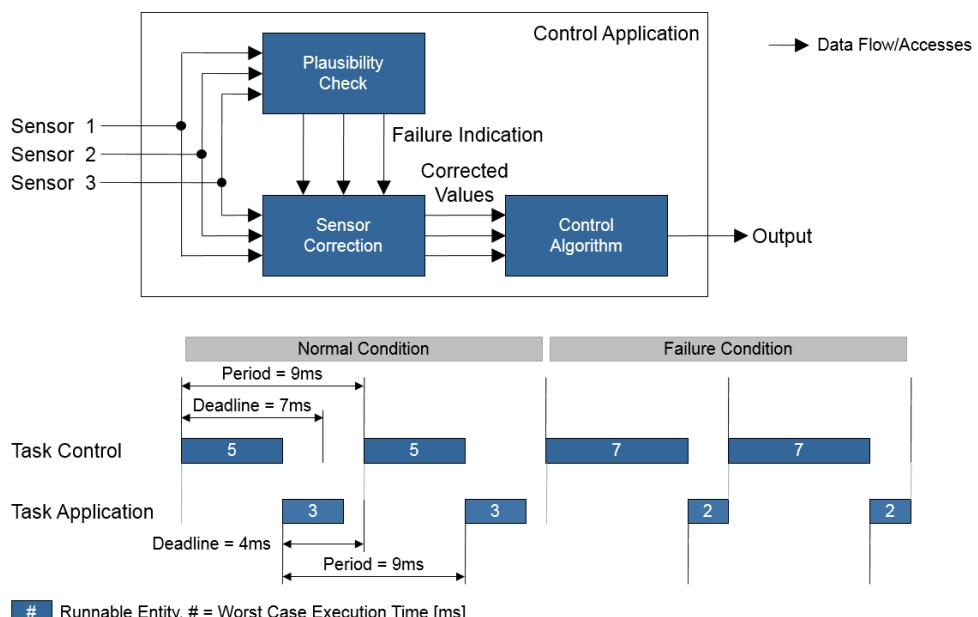


Figure 2.9: Generic fault tolerant control application

Assumed the application executing the three runnable entities is mapped to an ECU executing already runnable entities of another application. The runnable entities of these two applications are executed in the context of two tasks called Task Control and Task Application. The former task executes the runnable entities of the given fault tolerant application and the latter task executes the application already being mapped to the ECU and consisting of a number of runnable entities. Both tasks respectively applications are activated periodically every 9ms. The task Task Control has a deadline of 7ms and the task Task Application has a deadline of 4ms.

As mentioned above the first application operates under two different conditions: Under normal condition the execution time of the runnable entities does not exceed 5ms, and under failure condition the execution time of the runnable entities increases to 7ms due

to the additional time required for executing the algorithm to derive the value of the erroneous sensor.

A rough sketch of the task schedule containing both tasks is given in the lower part of Figure 2.9. On the left hand side the fault tolerant application is operating under normal conditions and on the right hand side this application is operating under failure condition. In this case — failure condition — the execution time 3ms of the task ask Application leads to a violation of the schedule, which means that the two applications cannot be integrated onto the same ECU unless both applications are mapped onto a more powerful ECU — resulting in a decrease of execution times. Another possibility would be that the application executed by the task Task Application provides capabilities to operate under degraded conditions resulting in a shorter execution time, for example 2ms.

However, in the cases described above timing constraints imposed on timing properties of SWCs and/or runnable entities depend on specific conditions a system shall operate properly.

TIMEX is capable of specifying timing constraints for specific conditions as shown in Figure 2.10. Such a timing constraint is valid if and only if the given timing condition holds.

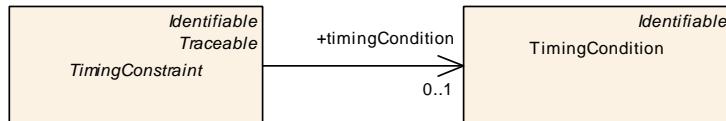


Figure 2.10: Conditional Timing Constraint

Since a [TimingConstraint](#) plays either the role of a timing requirement or timing guarantee the dependency on a condition can be specified on both types of timing information, namely timing constraint and timing property.

[TPS_TIMEX_00049] Purpose of [TimingCondition](#)

Upstream requirements: [RS_TIMEX_00011](#)

[The purpose of the [TimingCondition](#) is to describe a condition a timing constraint is depending on.]

[TPS_TIMEX_00050] Purpose of [TimingConditionFormula](#)

Upstream requirements: [RS_TIMEX_00011](#)

[The purpose of the [TimingConditionFormula](#) is to specify an expression describing a dependency on a specific condition.]

[TPS_TIMEX_00051] Purpose of [TimingExtensionResource](#)

Upstream requirements: [RS_TIMEX_00011](#)

[The purpose of the [TimingExtensionResource](#) is to subsume a number of reusable elements, like instance references to various timing relevant elements, that

are referenced from within a [TimingConditionFormula](#). These elements of a [TimingExtensionResource](#) are related to the context of the aggregating — *parent* — [TimingExtension](#).]

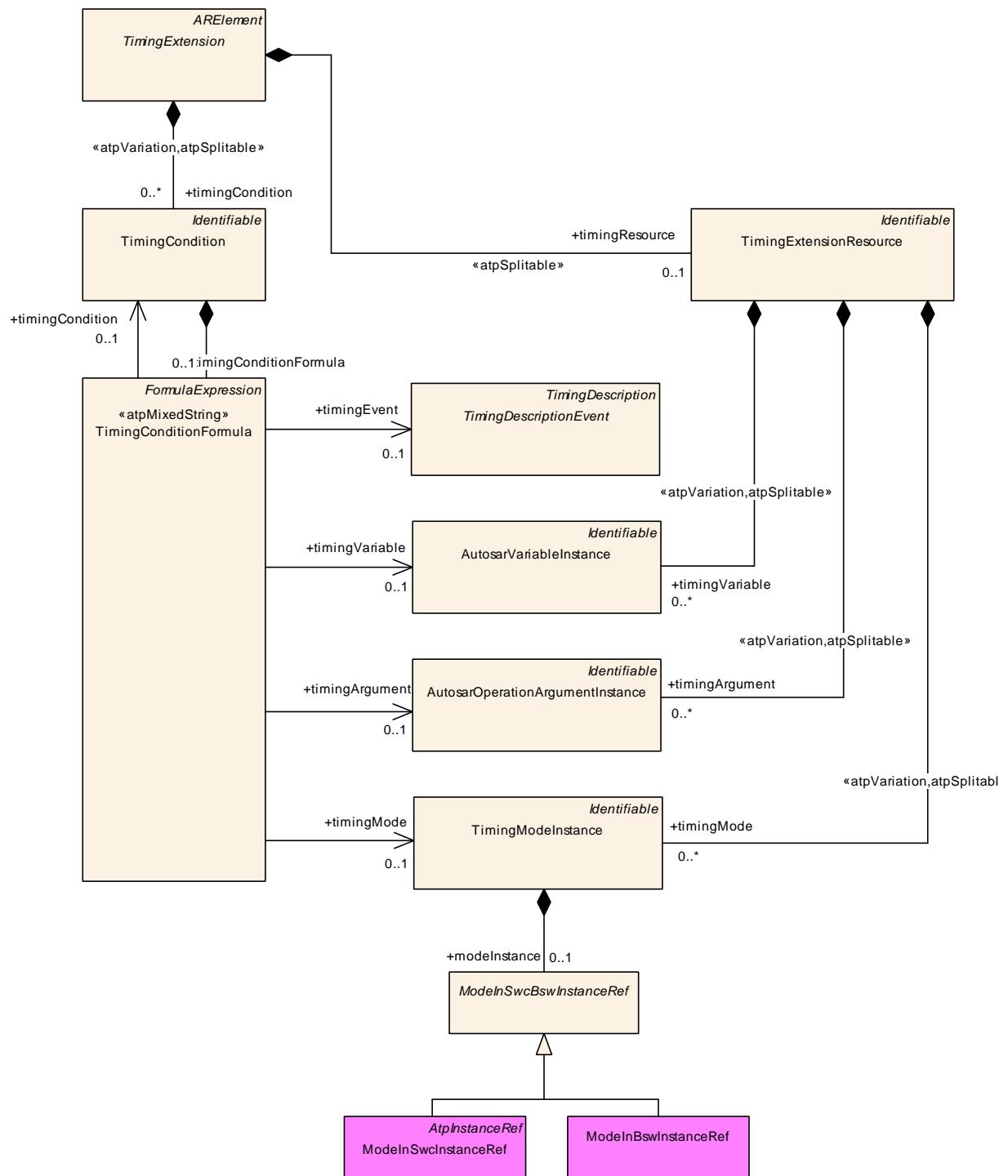


Figure 2.11: Conditional Timing

| Class | TimingCondition | | | |
|----------------------------|---|-------|------|---|
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingCondition | | | |
| Note | A TimingCondition describes a dependency on a specific condition. The element owns an expression which describes the timing condition dependency. | | | |
| Base | ARObject, Identifiable , MultilanguageReferrable , Referrable | | | |
| Aggregated by | TimingExtension.timingCondition | | | |
| Attribute | Type | Mult. | Kind | Note |
| timingCondition Formula | TimingCondition Formula | 0..1 | aggr | This is the expression describing the dependency on a specific condition. |

Table 2.7: TimingCondition

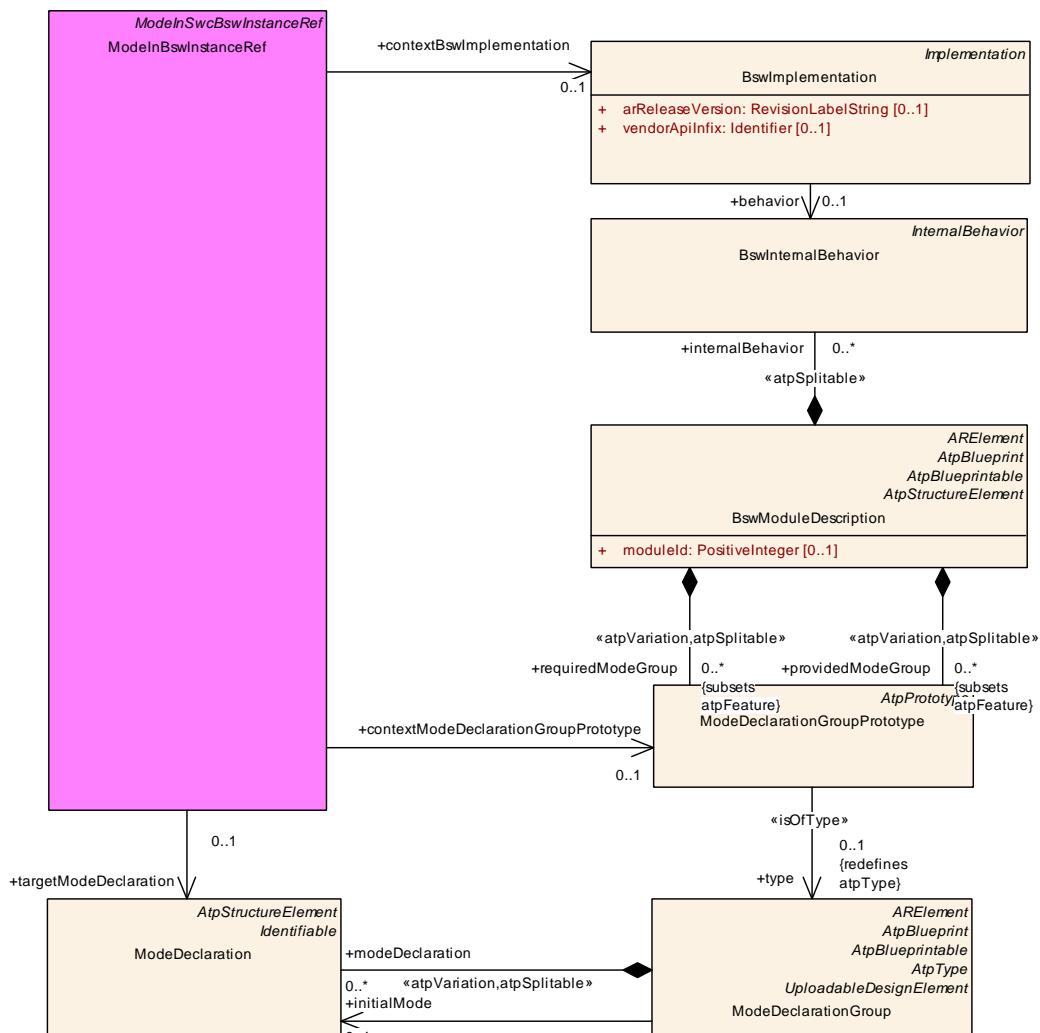
| Class | «atpMixedString» TimingConditionFormula | | | |
|-----------------|---|-------|------|--|
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingCondition | | | |
| Note | A TimingConditionFormula describes a specific dependency. The expression shall be a boolean expression addressing modes, variables, arguments, and/or events. | | | |
| Base | ARObject, FormulaExpression | | | |
| Aggregated by | TimingCondition.timingConditionFormula | | | |
| Attribute | Type | Mult. | Kind | Note |
| timingArgument | AutosarOperation ArgumentInstance | 0..1 | ref | This refers to an argument of an operation call. |
| timingCondition | TimingCondition | 0..1 | ref | This refers to a timing condition that is part of an expression describing the dependency on a specific condition. |
| timingEvent | TimingDescriptionEvent | 0..1 | ref | This refers to a timing event. |
| timingMode | TimingModeInstance | 0..1 | ref | This refers to a mode declaration. |
| timingVariable | AutosarVariable Instance | 0..1 | ref | This refers to a variable. |

Table 2.8: TimingConditionFormula

| Class | TimingExtensionResource | | | |
|----------------|---|-------|------|---|
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingCondition | | | |
| Note | A TimingExtensionResource provides the capability to contain instance references referred from within a timing condition formula. | | | |
| Base | ARObject, Identifiable , MultilanguageReferrable , Referrable | | | |
| Aggregated by | TimingExtension.timingResource | | | |
| Attribute | Type | Mult. | Kind | Note |
| timingArgument | AutosarOperation ArgumentInstance | * | aggr | <p>This refers to an instance reference of an argument of an operation call.</p> <p>Stereotypes: atpSplittable; atpVariation</p> <p>Tags: atp.Splitkey=timingArgument.shortName, timing Argument.variationPoint.shortLabel vh.latestBindingTime=postBuild</p> |



| Class | TimingExtensionResource | | | |
|----------------|-------------------------|---|------|---|
| timingMode | TimingModeInstance | * | aggr | <p>This refers to an instance reference of a mode declaration.</p> <p>Stereotypes: atpSplittable; atpVariation</p> <p>Tags: <code>atp.Splitkey=timingMode.shortName, timingMode.variationPoint.shortLabel</code> <code>vh.latestBindingTime=postBuild</code> </p> |
| timingVariable | AutosarVariableInstance | * | aggr | <p>This refers to an instance reference of a variable.</p> <p>Stereotypes: atpSplittable; atpVariation</p> <p>Tags: <code>atp.Splitkey=timingVariable.shortName, timingVariable.variationPoint.shortLabel</code> <code>vh.latestBindingTime=postBuild</code> </p> |

Table 2.9: TimingExtensionResource

Figure 2.12: Instance Reference Mode for Basic Software Module

Since the notion of "Type and Prototype" is not supported by the Basic Software Module Description Template, the element `ModeInBswInstanceRef` is not a specialization of

[AtpInstanceRef](#). Therefore, the directed association playing the role of "base" is not present.

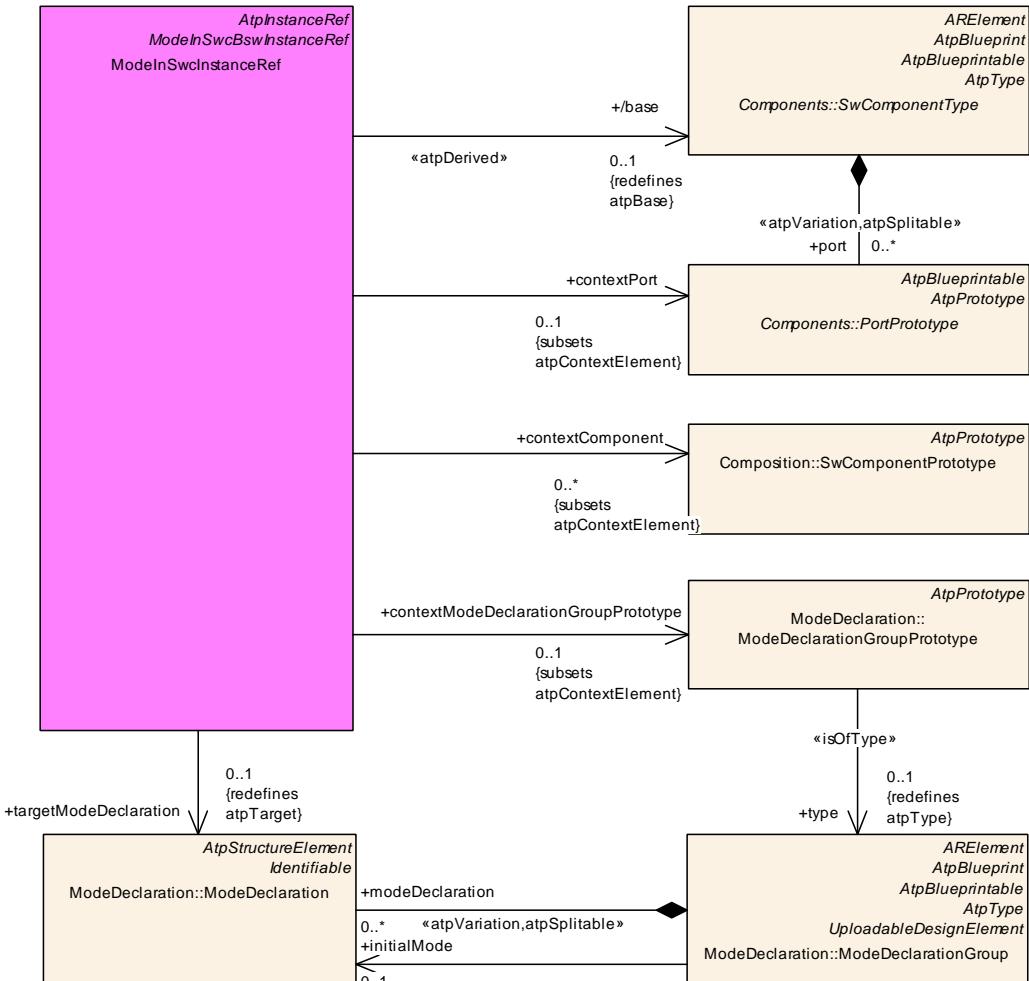


Figure 2.13: Instance Reference Mode for Software Component

| | | | | |
|----------------------|---|--------------|-------------|--|
| Class | TimingModelInstance | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingCondition | | | |
| Note | This class specifies the mode declaration to be checked in a specific instance of a mode declaration group. This is used in a timing condition formula as an operand of the unary timing function <code>TIMEX_mode</code> Active to check whether the mode declaration is active at the point in time this expression is evaluated. | | | |
| Base | <i>ARObject</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i> | | | |
| Aggregated by | TDEventOccurrenceExpression.mode, TimingExtensionResource.timingMode | | | |
| Attribute | Type | Mult. | Kind | Note |
| modelInstance | ModelInSwcBswInstanceRef | 0..1 | aggr | This refers to a specific mode declaration in the given context. |

Table 2.10: TimingModelInstance

| | | | | |
|--------------------------------------|--|--------------|-------------|--|
| Class | ModelInBswInstanceRef | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingCondition | | | |
| Note | Instance reference to be capable of referencing a specific ModeDeclaration of a ModeDeclarationGroup Prototype utilized in a BSW module. | | | |
| Base | <i>ARObject</i> , ModelInSwcBswInstanceRef | | | |
| Aggregated by | TimingModelInstance.modelInstance | | | |
| Attribute | Type | Mult. | Kind | Note |
| contextBswImplementation | BswImplementation | 0..1 | ref | Specifies the BSW implementation that manifests the context. Tags: xml.sequenceOffset=10 |
| contextModeDeclarationGroupPrototype | ModeDeclarationGroupPrototype | 0..1 | ref | Specifies the mode declaration group prototype that manifests the context. Tags: xml.sequenceOffset=20 |
| targetModeDeclaration | ModeDeclaration | 0..1 | ref | Specifies the specific mode declaration in the given context. Tags: xml.sequenceOffset=30 |

 Table 2.11: **ModelInBswInstanceRef**

[constr_6853] Existence of [ModelInBswInstanceRef.contextModeDeclarationGroupPrototype](#)

Imposition time: [IT_BswTd](#)

For each [ModelInBswInstanceRef](#), the reference to [ModeDeclarationGroupPrototype](#) in the role [contextModeDeclarationGroupPrototype](#) shall exist at least once.]

[constr_6854] Existence of [ModelInBswInstanceRef.targetModeDeclaration](#)

Imposition time: [IT_BswTd](#)

For each [ModelInBswInstanceRef](#), the reference to [ModeDeclaration](#) in the role [targetModeDeclaration](#) shall exist at least once.]

| | | | | |
|----------------------|---|--------------|-------------|--|
| Class | ModelInSwcInstanceRef | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingCondition | | | |
| Note | Instance reference to be capable of referencing a ModeDeclaration at a specific Mode Switch Port of a SW-C. | | | |
| Base | <i>ARObject</i> , AtpInstanceRef , ModelInSwcBswInstanceRef | | | |
| Aggregated by | TimingModelInstance.modelInstance | | | |
| Attribute | Type | Mult. | Kind | Note |
| base | SwComponentType | 0..1 | ref | Specifies the SW component representing the base of the context. Stereotypes: atpDerived Tags: xml.sequenceOffset=10 |
| contextComponent | SwComponentPrototype | * | ref | Specifies the SW component prototype representing the context. Tags: xml.sequenceOffset=20 |





| Class | ModeInSwcInstanceRef | | | |
|--|--------------------------------|------|-----|--|
| contextMode Declaration GroupPrototype | ModeDeclarationGroup Prototype | 0..1 | ref | Specifies the mode declaration group prototype that manifests the context. Tags: xml.sequenceOffset=40 |
| contextPort | PortPrototype | 0..1 | ref | Specifies the port prototype representing the context. Tags: xml.sequenceOffset=30 |
| targetMode Declaration | ModeDeclaration | 0..1 | ref | Specifies the specific mode declaration in the given context. Tags: xml.sequenceOffset=50 |

Table 2.12: ModeInSwcInstanceRef

[constr_6855] Existence of ModeInSwcInstanceRef.contextModeDeclarationGroupPrototype

Imposition time: IT_SwcTd

〔For each ModeInSwcInstanceRef, the reference to ModeDeclarationGroupPrototype in the role contextModeDeclarationGroupPrototype shall exist at least once.〕

[constr_6856] Existence of ModeInSwcInstanceRef.contextPort

Imposition time: IT_SwcTd

〔For each ModeInSwcInstanceRef, the reference to PortPrototype in the role contextPort shall exist at least once.〕

[constr_6857] Existence of ModeInSwcInstanceRef.targetModeDeclaration

Imposition time: IT_SwcTd

〔For each ModeInSwcInstanceRef, the reference to ModeDeclaration in the role targetModeDeclaration shall exist at least once.〕

[constr_6899] Existence of ModeInSwcInstanceRef.base

Imposition time: IT_SwcTd

〔For each ModeInSwcInstanceRef, the reference to SwComponentType in the role base shall exist at least once.〕

2.3 TimingDescription

The TimingDescription of a system, sub-system or SWC consists of observable TimingDescriptionEvents and their causal relationships (TimingDescriptionEventChain) as shown in Figure 2.14. These are detailed in the next sections.

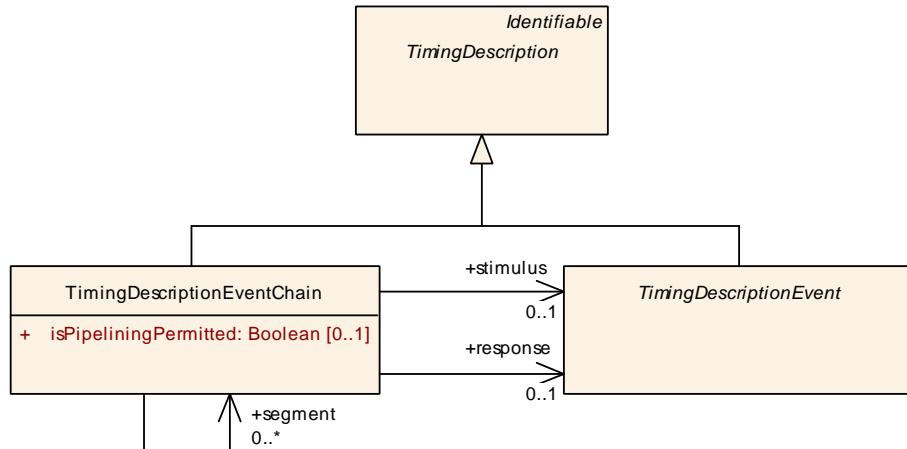


Figure 2.14: TimingDescription

2.3.1 TimingDescriptionEventChain

Event chains ([TimingDescriptionEventChain](#)s) specify a causal relationship between functionally dependent events ([TimingDescriptionEvent](#)s) and their temporal occurrences. Generally, the notion of a [TimingDescriptionEventChain](#) enables one to specify the relationship between two [TimingDescriptionEvent](#) points, for example: when a event *A* occurs then the event *B* occurs, or in other words, the event *B* occurs if and only if the event *A* occurred before.

In the context of a [TimingDescriptionEventChain](#) the event *A* has the role of the [stimulus](#) and the event *B* has the role of the [response](#). By means of an [TimingDescriptionEventChain](#), the correlation between a stimulation of a system and its corresponding response can be explicitly described and used as a formalized definition of the scope for [TimingConstraints](#).

Event chains can be hierarchical/nested, i.e. composed of further [TimingDescriptionEventChain](#)s and decomposed into further [TimingDescriptionEventChain](#)s — in both cases the nested [TimingDescriptionEventChain](#)s are named [segments](#).

[TPS_TIMEX_00002] Purpose of [TimingDescriptionEventChain](#)

Upstream requirements: [RS_TIMEX_00001](#), [RS_TIMEX_00004](#), [RS_TIMEX_00005](#), [RS_TIMEX_00009](#)

[The element [TimingDescriptionEventChain](#) is used to specify a causal relationship between timing description events and their occurrences during the runtime of a system.]

[TPS_TIMEX_00111] Semantics of [TimingDescriptionEventChain.stimulus](#)

Status: DRAFT

[A [stimulus](#) represents the starting point (or initial [TimingDescriptionEvent](#)) of a [TimingDescriptionEventChain](#)]

[TPS_TIMEX_00114] Semantics of [TimingDescriptionEventChain.response](#)

Status: DRAFT

[A [response](#) represents the end point (or final [TimingDescriptionEvent](#)) of a [TimingDescriptionEventChain](#)]

| | | | | |
|------------------------|--|--------------|-------------|---|
| Class | TimingDescriptionEventChain | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription | | | |
| Note | An event chain describes the causal order for a set of functionally dependent timing events. Each event chain has a well defined stimulus and response, which describe its start and end point. Furthermore, it can be hierarchically decomposed into an arbitrary number of sub-chains, so called <i>event chain segments</i> . | | | |
| Base | ARObject , Identifiable , MultilanguageReferrable , Referrable , TimingDescription | | | |
| Aggregated by | TimingExtension.timingDescription | | | |
| Attribute | Type | Mult. | Kind | Note |
| isPipelining Permitted | Boolean | 0..1 | attr | States whether the scheduled entities in an LET interval shall use pipelined execution or not i.e. "permitted pipelining property" If TRUE, then the scheduled entities must implement pipelining. If FALSE or undefined, no pipelining applies. Tags: atp.Status=draft |
| response | TimingDescriptionEvent | 0..1 | ref | The response event representing the point in time where the event chain is terminated. Tags: xml.sequenceOffset=20 |
| segment | TimingDescriptionEvent Chain | * | ref | A composed event chain consists of an arbitrary number of sub-chains. Tags: xml.sequenceOffset=30 |
| stimulus | TimingDescriptionEvent | 0..1 | ref | The stimulus event representing the point in time where the event chain is activated. Tags: xml.sequenceOffset=10 |

Table 2.13: [TimingDescriptionEventChain](#)

Depending on the value of the [categorys](#) of the [TimingDescriptionEventChain](#), it may be used in different use-cases.

[TPS_TIMEX_00110] Standardized [categorys](#) of [TimingDescriptionEventChain](#)

Status: DRAFT

[AUTOSAR standardizes the following [categorys](#) of [TimingDescriptionEventChain](#) and their semantics:

- undefined: as per STANDARD

- **STANDARD**: No specific semantics are imposed on the `TimingDescriptionEventChain`. It indicates the standard behavior.
- **LET_INTERVAL**: The `TimingDescriptionEventChain` represents an LET interval
- **SL LET_INTERVAL**: The `TimingDescriptionEventChain` represents a SL-LET interval

]

[constr_4515] Orthogonality of `stimulus` and `response` in a `TimingDescriptionEventChain`*Imposition time:* `IT_SubClasTeAss`

[The reference `TimingDescriptionEventChain.stimulus` and the reference `TimingDescriptionEventChain.response` shall not reference the same `TimingDescriptionEvent.shortName`.]

[constr_6906] Conformity of `stimulus` and `response` in a `TimingDescriptionEventChain`*Status:* DRAFT*Imposition time:* `IT_SubClasTdEvAss`

[The `TimingDescriptionEvent`s referenced in the roles `stimulus` and `response` shall be of the same sub-class (of `TimingDescriptionEvent`)]

[constr_6895] Existence of `TimingDescriptionEventChain.response`*Imposition time:* `IT_SubClasTdEvAss`

[For each `TimingDescriptionEventChain`, the reference in the role `response` shall exist]

[constr_6896] Existence of `TimingDescriptionEventChain.stimulus`*Imposition time:* `IT_SubClasTdEvAss`

[For each `TimingDescriptionEventChain`, the reference in the role `stimulus` shall exist]

[constr_6897] Existence of `TimingDescriptionEventChain.segment`*Imposition time:* `IT_SubClasTdEvAss`

[For each `TimingDescriptionEventChain`, the reference in the role `segment` shall exist at least once]

Figure 2.15 shows an event chain "End-to-End Timing" describing the causal dependency between "Sensor" and "Actuator". The sequence of event chain segments shows the details of "End-to-End Timing" according to the AUTOSAR timing views.

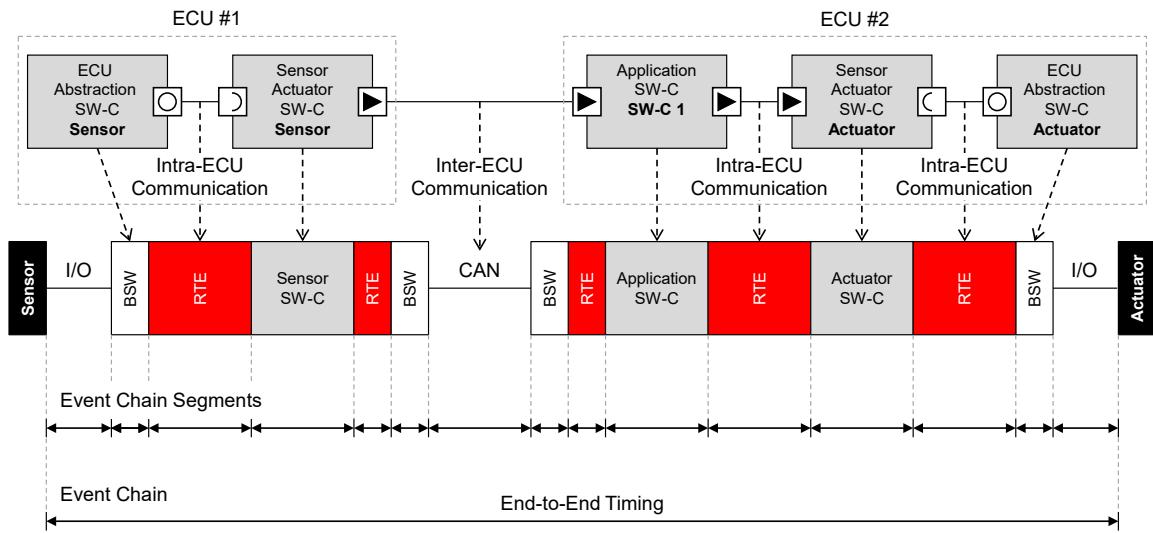


Figure 2.15: Example of an end-to-end event chain between sensor sampling and actuator access

2.3.1.1 Segments

[constr_4516] Completeness of a composed `TimingDescriptionEventChain`

Imposition time: `IT_SubClasTeAss`

〔If a `TimingDescriptionEventChain` has > 0 `segments`: after [constr_4518] is applied, there shall be at least one end-to-end path from the parental `TimingDescriptionEventChain.stimulus`, through the `segments`, to the parental `TimingDescriptionEventChain.response`.〕

[constr_6921] Disallow `TimingDescriptionEventChain` segmental circular-referencing

Status: DRAFT

Imposition time: `IT_SubClasTeAss`

〔A `TimingDescriptionEventChain.segment` shall never reference the (parent) `TimingDescriptionEventChain` in which it is referenced in the role `segment`.〕

[constr_4518] Specifying end-points of a composed `TimingDescriptionEventChain`

Imposition time: `IT_SubClasTeAss`

〔If a `TimingDescriptionEventChain` has > 0 `segments`: in *that* list of `segments`:

- at least one `segment.stimulus` shall reference the (parent) `TimingDescriptionEventChain.stimulus` in which it is referenced in the role `segment`
- at least one `segment.response` shall reference the (parent) `TimingDescriptionEventChain.response` in which it is referenced in the role `segment`

]

2.3.1.2 Approach

The following subsections describe how to structure event chains for systems. Depending on the pre-conditions two different approaches can be distinguished: top-down (decomposition) and bottom-up (composition).

The decomposition respectively composition of event chains can be performed according to the software component hierarchy, but does not necessarily have to follow this hierarchy. The primary purpose is to increase respectively decrease granularity of the timing descriptions.

Note that event chains are used in all AUTOSAR timing views and any composition and decomposition of event chains can be done across various AUTOSAR timing views.

2.3.1.2.1 Decomposition

In a first step the time critical path in the system is identified. This means that a causal relationship between a stimulus event and response event is described by an event chain. For this event chain a timing constraint is specified describing the time budget. The second step is to decompose this event chain into event chain segments which implies that the given time budget gets split — decomposed —, too.

Since event chain segments are event chains as well, these event chain segments can be subject to further decomposition.

Figure 2.16 shows a time critical path between the event "requesting the brake pedal position" (*Stimulus*) and the event "making available the determined vehicle speed" (*Response*). This event chain (*EC*) is subject to a timing constraint, namely a `LatencyTimingConstraint`, and is budgeted accordingly. For example, the time budget for the event chain *EC* is constrained by a maximum latency of 2 ms.

In subsequent steps of the development and with deeper knowledge about the system's dynamics, this event chain and its time budget can be split across the system's components. This results in the event chain segments *EC1*, *EC2* and *EC3* and their appropriate time budgets. The sum of these time budgets shall not exceed the given time budget of 2 ms.

2.3.1.2.2 Composition

In the first step the system is build up based on available software components including timing descriptions. In the second step available event chains are connected with each other. This results in a sequence of event chains where the response event of one event chain has the role of the stimulus event of the subsequent event chain. In the third step, a high-level event chain is specified based on a sequence of available event chains which play the role of event chain *segments*. For this high-level event chain a time budget shall be specified. Finally, the aggregated time budget needs to be assessed if acceptable which means that the aggregated time budget shall be equal or less than the time budget of the high-level event chain.

Figure 2.16 shows the connected event chains EC_1 , EC_2 and EC_3 . For each event chain a time budget, using a [LatencyTimingConstraint](#), is specified: The time budget of event chain EC_1 is 0.5 ms, of event chain EC_2 is 0.6 ms and of event chain EC_3 is 0.7 ms. The high-level event chain EC is a composition of the event chains EC_1 , EC_2 and EC_3 . The stimulus event of the high-level event chain is the event "requesting the brake pedal position" (*Stimulus*) and the response event of the high-level event chain is the event "making available the determined vehicle speed" (*Response*). Eventually, a time budget is assigned to the high-level event chain using a [LatencyTimingConstraint](#), for example 2 ms. This value is consistent with the aggregated time budget of the event chain segments ($0.5\text{ ms} + 0.6\text{ ms} + 0.7\text{ ms} = 1.8\text{ ms}$).

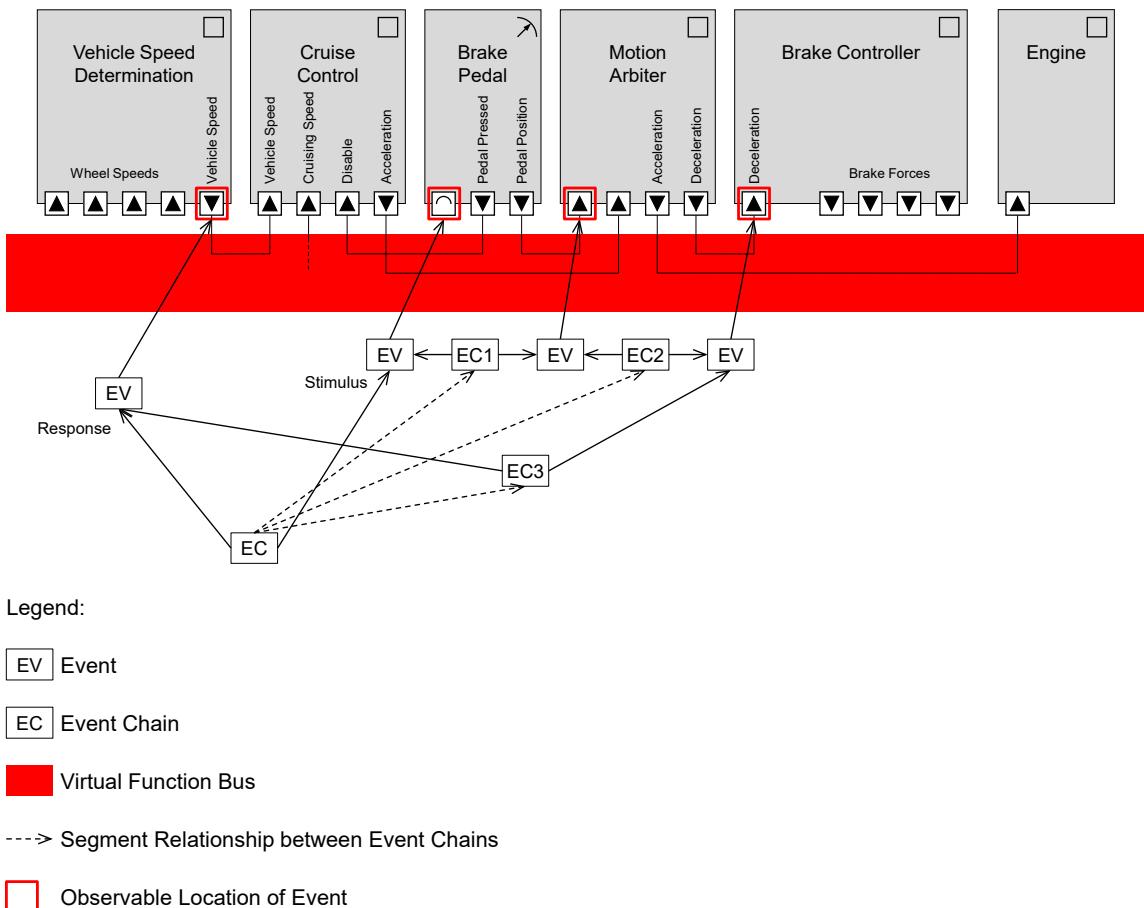


Figure 2.16: Example of a composed and decomposed event chain

2.3.1.3 Patterns

A sequence or hierarchy of event chains can form complex structures. However, if one of the aforementioned approaches is correctly followed then there is only a handful of patterns applicable. These patterns are introduced in the following with a simple example.

2.3.1.3.1 Sequence

The most frequently used pattern is the sequence of events. Such a sequence describes a succession of causally related events without an alternative path.

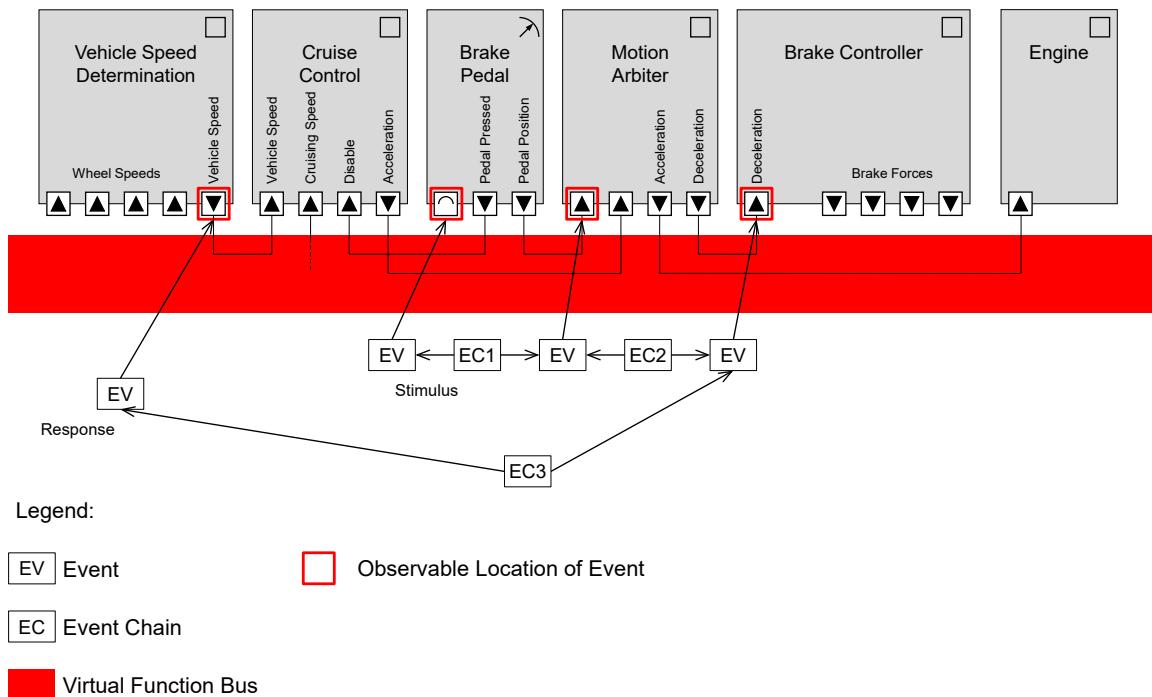


Figure 2.17: Example of the "Sequence" pattern

An example for this pattern is depicted in Figure 2.17. The event chains *EC1* through *EC3* define a causal relationship of events observed at a port of the SWC called *Brake Pedal* and a port of the SWC called *Vehicle Speed Determination*.

2.3.1.3.2 Fork

The "Fork" pattern describes the constellation where several event chains have one common stimulus event and different response events.

The pattern is illustrated in Figure 2.18, which shows a path that forks because the SWC *Brake Controller* calculates the brake force value for each wheel (*EC5* through *EC8*).

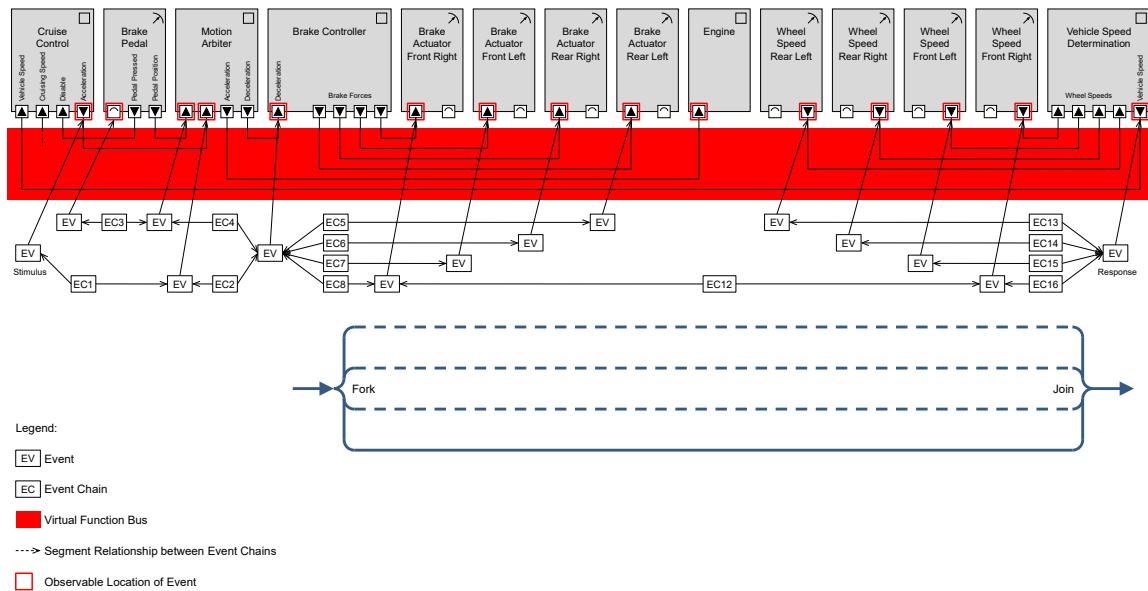


Figure 2.18: Example of the "Fork" and "Join" pattern

2.3.1.3.3 Join

The "Join" pattern describes the constellation where several event chains have one common response event and different stimulus events.

The pattern is illustrated in Figure 2.18 which shows a path that joins because the SWC *Vehicle Speed Determination* aggregates the wheel speed values from individual wheels (EC13 through EC16).

2.3.1.3.4 Alternative

The "Alternative" pattern describes the constellation where more than one path between a stimulus and response event exists. This implies that at least one "Fork" is followed by at least one "Join".

The pattern is illustrated in Figure 2.19 which shows that an event observed at a required port of the SWC *Motion Arbitrator* leads to an occurrence of an event either at the port called *Deceleration* of the SWC *Brake Controller*, or at the port called *Acceleration* of the SW-C *Engine*. These alternative causal relationships are described by the event chains *EC2* and *EC4* in this figure. In either case, the deceleration or acceleration of the vehicle leads to the occurrence of an event at the provided port called *Vehicle Speed* of the SW-C *Vehicle Speed Determination* reporting the vehicle's speed. These alternative causal relationships are described by the event chains *EC3* and *EC5* which both reference the same response event. To fulfill the overall event chain, only one of the alternative paths shall have been occurred.

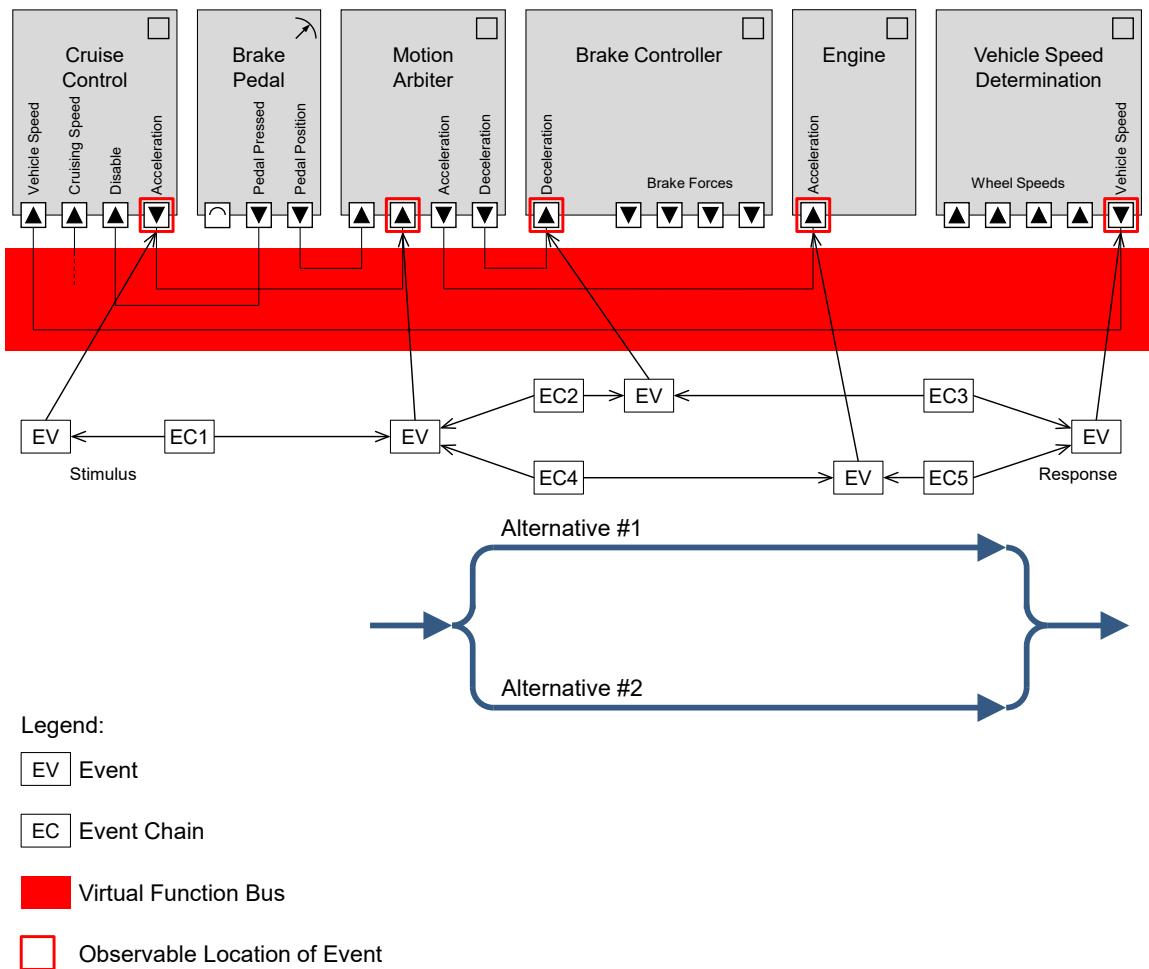


Figure 2.19: Example of the "Alternative" pattern

2.3.1.3.5 Cycle

The "Cycle" pattern describes the constellation where a path from the response event of an event chain leads to the stimulus of this event chain.

The pattern is illustrated in Figure 2.20 which shows three event chains *EC8*, *EC12* and *EC17* forming a cycle. The stimulus event of event chain *EC8* is the response event of event chain *EC17*; and the response event of event chain *EC12* is the stimulus event of event chain *EC17*. Event chain *EC8* and *EC12* reference the same event in different roles, namely response event from event chain *EC8* perspective and stimulus event from the event chain *EC12* perspective.

Note that an event chain referencing the same event for its stimulus and its response is forbidden according to the constraint [constr_4515]. As a consequence a cycle consists of at least two event chains.

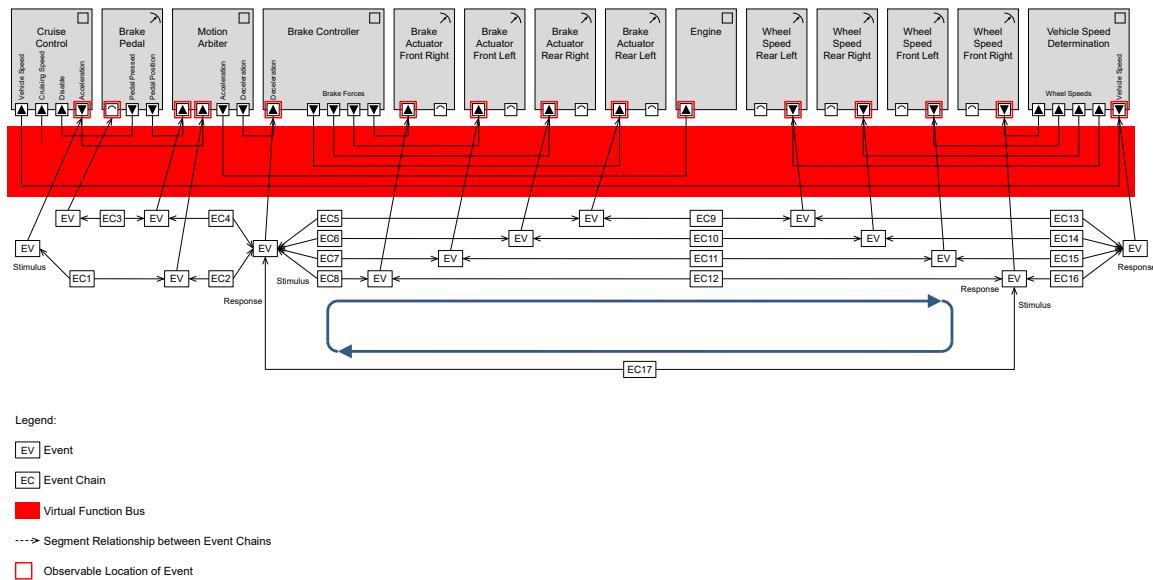


Figure 2.20: Example of the "Cycle" pattern

2.3.2 TimingDescriptionEvent

TimingDescriptionEvents refer to locations in systems at which the *occurrences* of events are observed at runtime. TIMEX defines a set of predefined TimingDescriptionEvent types for use in differing *observable locations* and further in differing Timing Views (2.1).

[TPS_TIMEX_00001] Purpose of TimingDescriptionEvent

Upstream requirements: RS_TIMEX_00001

[The element `TimingDescriptionEvent` and its specializations are used to describe the occurrences of a timing event observed at a specific location in a system during runtime respectively the operation of the system.]

For example, this can be the start of a `RunnableEntity` or storing a frame in the hardware buffer of a communication controller.

An overview of the different event types is given in Figure 2.21. These are described in more detail in the following sub-sections.

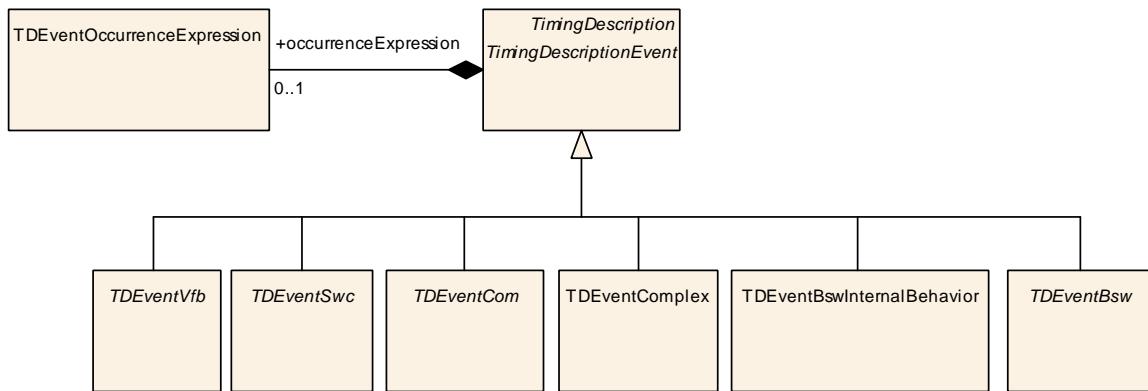


Figure 2.21: Overview of the different types of timing events

Depending on the value of the [category](#) of the [TimingDescriptionEvent](#), it may be used in different use-cases.

[TPS_TIMEX_00056] Standardized [category](#)s of [TimingDescriptionEvent](#) in Classic Platform [AUTOSAR standardizes the following [category](#)s of [TimingDescriptionEvent](#) and their semantics:

- [undefined](#): as per [STANDARD](#)
- [STANDARD](#): No specific semantics are imposed on the [TimingDescriptionEvent](#). It indicates the standard behavior.
- [LET_RELEASE](#): The [TimingDescriptionEvent](#) represents the release/start point of an LET interval
- [LET_TERMINATE](#): The [TimingDescriptionEvent](#) represents the termination/end point of an LET interval
- [DISPATCH_ENTRY_POINT](#): The [TimingDescriptionEvent](#) marks the *dispatcher entry point* of a timing description event chain describing a dispatcher associated with a software cluster
- [DISPATCH_EXIT_POINT](#): The [TimingDescriptionEvent](#) marks the *dispatcher exit point* of a timing description event chain describing a dispatcher associated with a software cluster

]

[constr_4559] Restriction of [TimingDescriptionEvent.category](#)

Imposition time: [IT_SubClasTdEvAss](#)

[Any [TimingDescriptionEvent.category](#) not in the list in [\[TPS_TIMEX_00056\]](#) shall be ignored.]

Also note that information regarding the occurrence of a [TimingDescriptionEvent](#) is described separately in [2.4.1.2](#).

2.3.2.1 TDEventVfb

[TPS_TIMEX_00016] Purpose of TDEventVfb

Upstream requirements: [RS_TIMEX_00001](#)

[The element [TDEventVfb](#) and its specializations are used to describe the occurrences of an event which are observed at a specific location in the VFB view.]

Events related to the VFB can be used during the specification of:

- [VfbTiming 2.1.1](#)
- [SystemTiming 2.1.3](#)
- [SwcTiming 2.1.2](#)
- [EcuTiming 2.1.5](#)

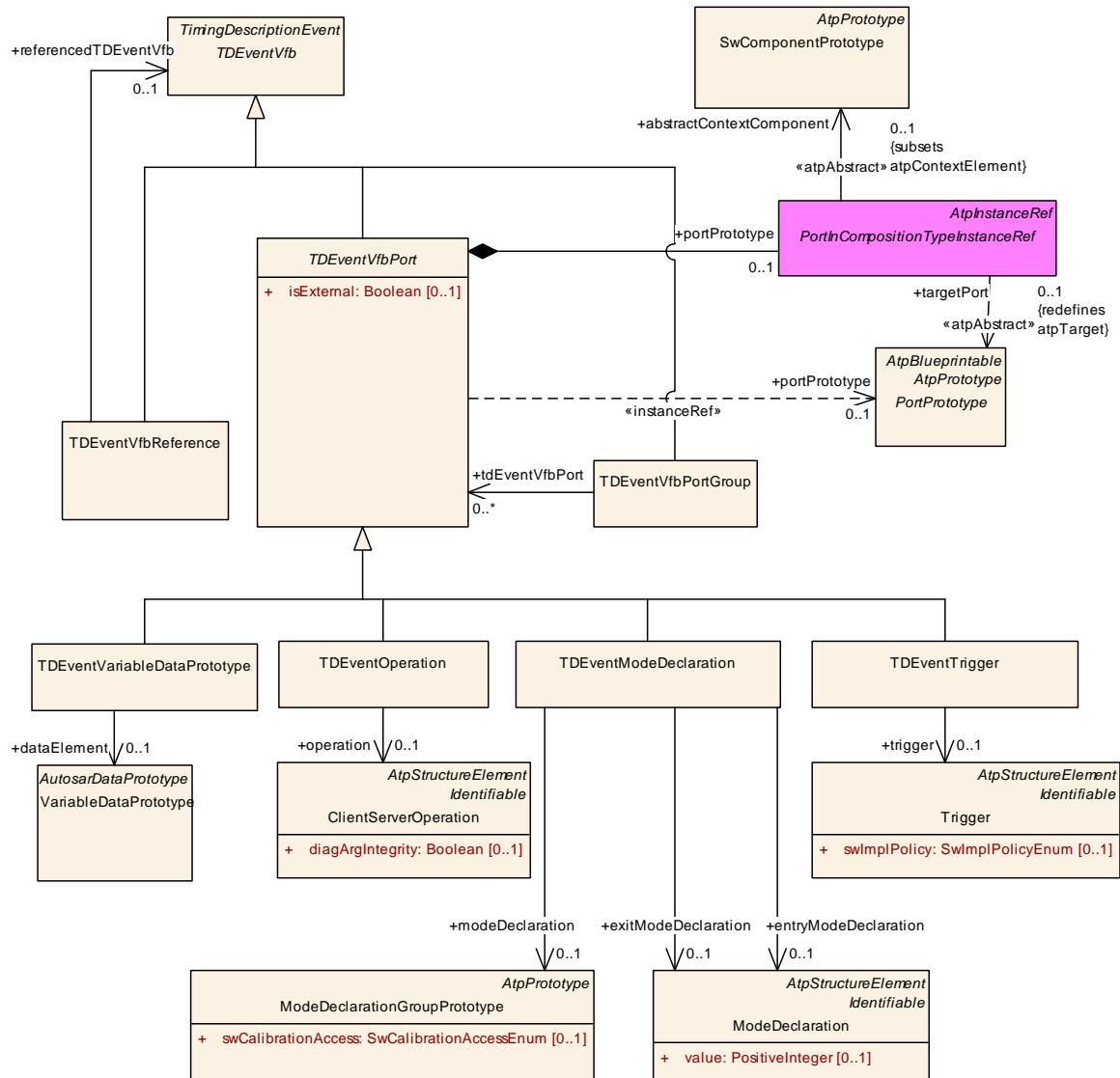


Figure 2.22: VFB events

| | | | | |
|----------------------|---|--------------|-------------|--|
| Class | <i>TDEventVfb</i> (abstract) | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescriptionEvents::TDEventVfb | | | |
| Note | A <i>TimingDescriptionEvent</i> occurring on a Virtual Functional Bus (VFB) <i>PortPrototype</i> . | | | |
| Base | ARObject, <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i> , <i>TimingDescription</i> , <i>TimingDescriptionEvent</i> | | | |
| Subclasses | <i>TDEventVfbPort</i> , <i>TDEventVfbPortGroup</i> , <i>TDEventVfbReference</i> | | | |
| Aggregated by | <i>TimingExtension.timingDescription</i> | | | |
| Attribute | Type | Mult. | Kind | Note |
| component | <i>SwComponentPrototype</i> | 0..1 | iref | The context for the scope of this timing event. InstanceRef implemented by: <i>ComponentInCompositionInstanceRef</i> |

 Table 2.14: **TDEventVfb**

2.3.2.1.1 TDEventVfbReference

[TPS_TIMEX_00043] Purpose of **TDEventVfbReference**

Upstream requirements: [RS_TIMEX_00001](#), [RS_TIMEX_00019](#)

〔The element **TDEventVfbReference** is used to reference timing description events already specified in other timing views. In other words, it enables one to re-use existing timing models.〕

| | | | | |
|-----------------------|---|--------------|-------------|--|
| Class | TDEventVfbReference | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescriptionEvents::TDEventVfb | | | |
| Note | Reference to "other" <i>TimingDescriptionEvent</i> s. These other <i>TimingDescriptionEvent</i> s may be specified in other views and re-used in this view. | | | |
| Base | ARObject, <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i> , <i>TDEventVfb</i> , <i>TimingDescription</i> , <i>TimingDescriptionEvent</i> | | | |
| Aggregated by | <i>TimingExtension.timingDescription</i> | | | |
| Attribute | Type | Mult. | Kind | Note |
| referenced TDEventVfb | <i>TDEventVfb</i> | 0..1 | ref | The referenced timing description event. |

 Table 2.15: **TDEventVfbReference**

[constr_6886] Existence of **TDEventVfbReference.referencedTDEventVfb**

Imposition time: [IT_VfbTd](#)

〔For each **TDEventVfbReference**, the reference to **TDEventVfb** in the role **referencedTDEventVfb** shall exist〕

2.3.2.1.2 TDEventVfbPort

A [TDEventVfbPort](#) describes a [TimingDescriptionEvent](#) on the level of a [PortPrototype](#). Since [PortPrototypes](#) are typed by different types of [PortInterface](#)s whose usage is further broken down into the type of Operation/Data on that [PortInterface](#), the sub-classing of [TDEventVfbPort](#) has been further divided into similar usages:

- [TDEventVariableDataPrototype](#): the sending/receiving of a [dataElement](#)
- [TDEventOperation](#): making/receiving a function call, sending/receiving a response
- [TDEventTrigger](#): the sending/receiving of a [Trigger](#)
- [TDEventModeDeclaration](#): mode switching

Due to instantiation principles, the referenced elements from these [TDEventVfbPort](#)s, may be multiply instantiated, thus a [TDEventVfbPort](#) always needs to refer to the SWC context in the role [TDEventVfbPort.portPrototype](#). The meta-class [TDEventVfbPortGroup](#) is used to provide a grouping element for [TDEventVfbPort](#)s. This permits referencing a group of [TDEventVfbPort](#)s in a [TimingDescriptionEventChain.stimulus](#) or [TimingDescriptionEventChain.response](#), e.g. a group of [TDEventVariableDataPrototypes](#).

To permit some flexibility of referencing [PortPrototypes](#) on a system level (i.e. on heterogenous platform deployments), there are currently no general constraints that the grouped [PortPrototypes](#) must be all from the same SWC. This may be overridden for specific usages of a [TDEventVfbPortGroup](#) (e.g. SL-LET) in future. The usage is fully dependent on the multiplicity patterns for the type of [PortInterface](#) in which, after the multiplicity rules for the type of [PortInterface](#) see [6], [5] are applied to the [stimulus](#) or [response](#), permits a multiplicity > 1 for a [TDEventVfbPort](#).

[TPS_TIMEX_00042] Purpose of TDEventVfbPort

Upstream requirements: [RS_TIMEX_00001](#), [RS_TIMEX_00019](#)

[The element [TDEventVfbPort](#) and its specializations are used to describe the occurrences of an event which are observed at a specific location in the VFB view.]

| | |
|----------------------|--|
| Class | TDEventVfbPort (abstract) |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescriptionEvents::TDEventVfb |
| Note | A TimingDescriptionEvent occurring on a PortPrototype . |
| Base | ARObject , Identifiable , MultilanguageReferrable , Referrable , TDEventVfb , TimingDescription , TimingDescriptionEvent |
| Subclasses | TDEventModeDeclaration , TDEventOperation , TDEventTrigger , TDEventVariableDataPrototype |
| Aggregated by | TimingExtension.timingDescription |





| Class | <i>TDEventVfbPort</i> (abstract) | | | |
|-------------------------|----------------------------------|-------|------|---|
| Attribute | Type | Mult. | Kind | Note |
| isExternal | Boolean | 0..1 | attr | This attribute is used to refer to external events that are related to hardware I/O, like physical sensors and actuators, at Virtual Functional Bus (VFB) level. |
| port | PortPrototype | 0..1 | ref | port on which the TimingEvent shall apply Tags: atp.Status=obsolete |
| portPrototype | PortPrototype | 0..1 | iref | <i>PortPrototype</i> on which the TimingEvent occurs Tags: atp.Status=draft xml.typeElement=true InstanceRef implemented by: PortInCompositionType InstanceRef |
| portPrototype Blueprint | PortPrototypeBlueprint | 0..1 | ref | port on which the TimingEvent shall apply (in the context of an AUTOSAR blueprint) |

Table 2.16: *TDEventVfbPort*

[constr_6885] Existence of *TDEventVfbPort.isExternal*

Imposition time: [IT_VfbTd](#)

〔For each *TDEventVfbPort*, the attribute *isExternal* shall exist〕

In order to support the description of timing events for hardware I/O already at VFB-level (e.g. in order to refer to the point in time where data is generated by a physical sensor) without having the need to specify the concrete sensor hardware, it is necessary to specify the attribute *isExternal*.

If for a timing event of type *TDEventVfbPort* the attribute is set to "TRUE", then the timing event refers to the point in time where the data is generated/processed by the corresponding hardware I/O.

If the attribute is set to "FALSE", then the timing event refers to the point in time where the data enters or leaves the respective port of the component at VFB-level.

2.3.2.1.2.1 *TDEventVariableDataPrototype*

[TPS_TIMEX_00017] *TDEventVariableDataPrototype* specifies events observable at sender/receiver ports

Upstream requirements: [RS_TIMEX_00001](#)

〔The element *TDEventVariableDataPrototype* is used to specify events, namely the receipt and sending of variable data prototypes, observable at required and provided sender/receiver ports.〕

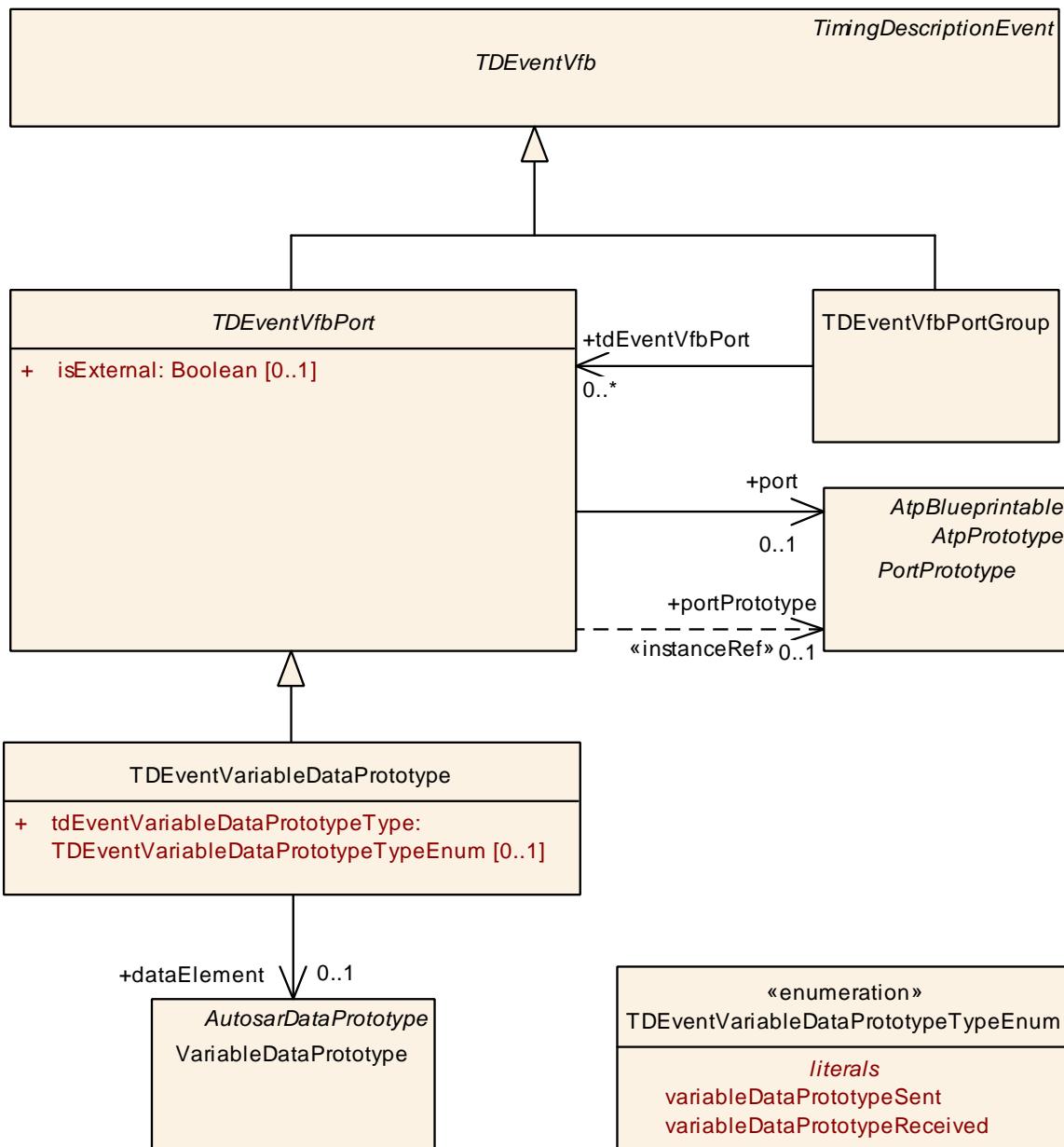


Figure 2.23: Variable Data Prototype

| | | | | |
|----------------------|---|--------------|-------------|---|
| Class | TDEventVariableDataPrototype | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescriptionEvents::TDEventVfb | | | |
| Note | A TimingDescriptionEvent triggered by the sending/receiving of a VariableDataPrototype in a SenderReceiverInterface on VFB level. | | | |
| Base | ARObject , Identifiable , MultilanguageReferrable , Referrable , TDEventVfb , TDEventVfbPort , TimingDescription , TimingDescriptionEvent | | | |
| Aggregated by | TimingExtension.timingDescription | | | |
| Attribute | Type | Mult. | Kind | Note |
| dataElement | VariableDataPrototype | 0..1 | ref | The referenced VariableDataPrototype from a SenderReceiverInterface . |



| Class | TDEventVariableDataPrototype | | | |
|----------------------------------|--------------------------------------|------|------|---|
| tdEventVariableDataPrototypeType | TDEventVariableDataPrototypeTypeEnum | 0..1 | attr | The specific type of this timing event. |

Table 2.17: TDEventVariableDataPrototype

[constr_6887] Existence of `TDEventVariableDataPrototype.tdEventVariableDataPrototypeType`

Imposition time: `IT_VfbTd`

〔For each `TDEventVariableDataPrototype`, the attribute `tdEventVariableDataPrototypeType` shall exist〕

[constr_6888] Existence of `TDEventVariableDataPrototype.dataElement`

Imposition time: `IT_VfbTd`

〔For each `TDEventVariableDataPrototype`, the reference to `VariableDataPrototype` in the role `dataElement`〕

| | |
|-------------------------------|---|
| Enumeration | <code>TDEventVariableDataPrototypeTypeEnum</code> |
| Package | <code>M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescriptionEvents::TDEventVfb</code> |
| Note | This is used to describe the specific event type of a <code>TDEventVariableDataPrototype</code> |
| Aggregated by | <code>TDEventVariableDataPrototype.tdEventVariableDataPrototypeType</code> |
| Literal | Description |
| variableDataPrototypeReceived | A point in time where the referenced variable data prototype has been successfully transmitted and is available in the related communication buffer (of the RTE) for the receiving SWC. Tags: <code>atp.EnumerationLiteralIndex=0</code> |
| variableDataPrototypeSent | A point in time where the referenced variable data prototype has been successfully sent out by the sending SWC, so that it is available in the related communication buffer (of the RTE) for transmission. Tags: <code>atp.EnumerationLiteralIndex=1</code> |

Table 2.18: TDEventVariableDataPrototypeTypeEnum

2.3.2.1.2.2 TDEventOperation

[TPS_TIMEX_00018] `TDEventOperation` specifies events observable at client/server ports.

Upstream requirements: `RS_TIMEX_00001`

〔The element `TDEventOperation` is used to specify events, namely the invocation of operations and their completion, observable at required and provided client/server ports.〕

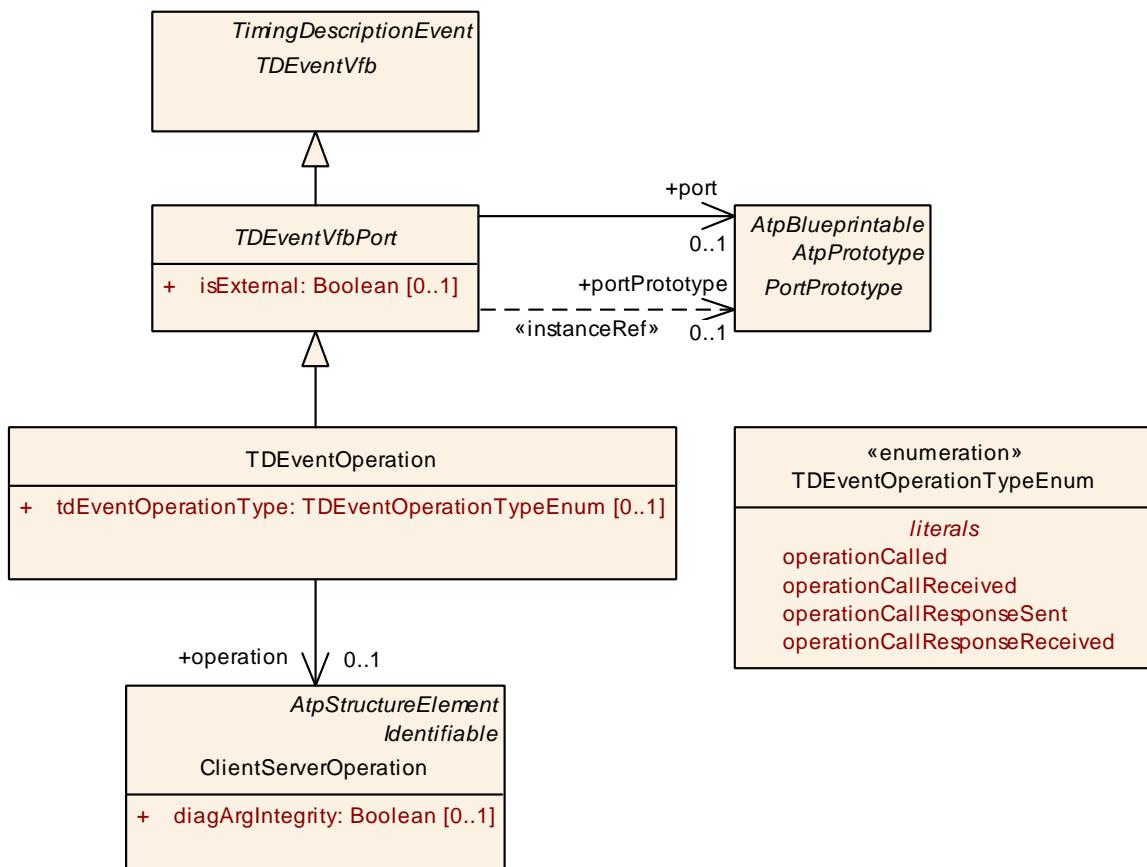


Figure 2.24: Operation

| Class | TDEventOperation | | | |
|--------------------|---|-------|------|---|
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescription | | | |
| Events::TDEventVfb | | | | |
| Note | A TimingDescriptionEvent triggered by the sending/receiving of a ClientServerOperation in a ClientServerInterface on VFB level. | | | |
| Base | ARObject , Identifiable , MultilanguageReferrable , Referrable , TDEventVfb , TDEventVfbPort , TimingDescription , TimingDescriptionEvent | | | |
| Aggregated by | TimingExtension.timingDescription | | | |
| Attribute | Type | Mult. | Kind | Note |
| operation | ClientServerOperation | 0..1 | ref | The referenced ClientServerOperation from a ClientServerInterface . |
| tdEvent | TDEventOperationType | 0..1 | attr | The specific type of this timing event. |
| OperationType | Enum | | | |

Table 2.19: TDEventOperation

[constr 6889] Existence of `TDEventOperation.tdEventOperationType`

Imposition time: $IT = Vfb/Td$

For each `TDEventOperation`, the attribute `tdEventOperationType` shall exist |

[constr_6890] Existence of `TDEventOperation.operation`

Imposition time: `IT_VfbTd`

〔For each `TDEventOperation`, the reference to `ClientServerOperation` in the role `operation` shall exist〕

| | |
|-------------------------------|---|
| Enumeration | <code>TDEventOperationTypeEnum</code> |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescriptionEvents::TDEventVfb |
| Note | This is used to describe the specific event type of a <code>TDEventOperation</code> . |
| Aggregated by | <code>TDEventOperation.idEventOperationType</code> |
| Literal | Description |
| operationCalled | A point in time where the referenced operation is called by the client SWC. Tags: <code>atp.EnumerationLiteralIndex=0</code> |
| operationCallReceived | A point in time where the call of the referenced operation is received by the server SWC. Tags: <code>atp.EnumerationLiteralIndex=1</code> |
| operationCallResponseReceived | A point in time where the client SWC has received the response of the referenced operation call. Tags: <code>atp.EnumerationLiteralIndex=2</code> |
| operationCallResponseSent | A point in time where the server SWC has terminated with the execution of the referenced operation, and has sent out a response. Tags: <code>atp.EnumerationLiteralIndex=3</code> |

Table 2.20: `TDEventOperationTypeEnum`

2.3.2.1.2.3 `TDEventModeDeclaration`

[TPS_TIMEX_00019] `TDEventModeDeclaration` specifies events observable at mode ports.

Upstream requirements: `RS_TIMEX_00001`

〔The element `TDEventModeDeclaration` is used to specify events, namely initiation and propagation of mode changes, observable at required and provided mode ports.〕

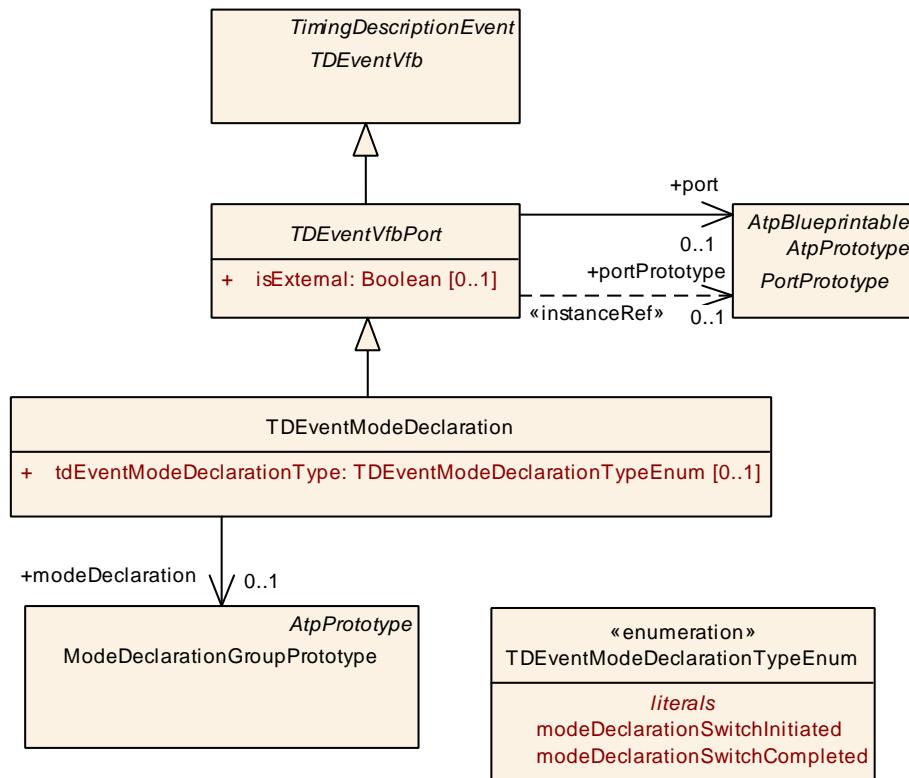


Figure 2.25: Mode Declaration

| | | | | |
|-----------------------------|---|--------------|-------------|--|
| Class | TDEventModeDeclaration | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescriptionEvents::TDEventVfb | | | |
| Note | A <code>TimingDescriptionEvent</code> triggered by a mode switch in a <code>ModeSwitchInterface</code> on VFB level. | | | |
| Base | <code>ARObject</code> , <code>Identifiable</code> , <code>MultilanguageReferrable</code> , <code>Referrable</code> , <code>TDEventVfb</code> , <code>TDEventVfbPort</code> , <code>TimingDescription</code> , <code>TimingDescriptionEvent</code> | | | |
| Aggregated by | <code>TimingExtension.timingDescription</code> | | | |
| Attribute | Type | Mult. | Kind | Note |
| entryMode Declaration | <code>ModeDeclaration</code> | 0..1 | ref | Optional parameter which refines the scope of the <code>TDEventModeDeclaration</code> . If the parameter is set, the event occurs only if the mode declaration group prototype instance shall enter into the referenced <code>ModeDeclaration</code> . |
| exitMode Declaration | <code>ModeDeclaration</code> | 0..1 | ref | Optional parameter which refines the scope of the <code>TDEventModeDeclaration</code> . If the parameter is set, the event occurs only if the mode declaration group prototype instance shall exit from the referenced <code>ModeDeclaration</code> . |
| mode Declaration | <code>ModeDeclarationGroupPrototype</code> | 0..1 | ref | The referenced <code>ModeDeclarationGroupPrototype</code> from a{ <code>ModeSwitchInterface</code> }. |
| tdEventMode DeclarationType | <code>TDEventModeDeclarationTypeEnum</code> | 0..1 | attr | The specific type of this timing event. |

Table 2.21: TDEventModeDeclaration

[constr_6891] Existence of `TDEventModeDeclaration.tdEventModeDeclarationType`

Imposition time: IT_VfbTd

〔For each `TDEventModeDeclaration`, the attribute `tdEventModeDeclarationType` shall exist〕

[constr_6892] Existence of `TDEventModeDeclaration.modeDeclaration`

Imposition time: IT_VfbTd

〔For each `TDEventModeDeclaration`, the reference to `ModeDeclarationGroupPrototype` in the role `modeDeclaration` shall exist〕

| <i>Enumeration</i> | <code>TDEventModeDeclarationTypeEnum</code> |
|------------------------------------|---|
| <i>Package</i> | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescriptionEvents::TDEventVfb |
| <i>Note</i> | This is used to describe the specific event type of a <code>TDEventModeDeclaration</code> |
| <i>Aggregated by</i> | <code>TDEventModeDeclaration.tdEventModeDeclarationType</code> |
| <i>Literal</i> | <i>Description</i> |
| modeDeclaration SwitchCompleted | A point in time where the switch to the associated <code>ModeDeclarationGroupPrototype</code> has been completed. Tags: atp.EnumerationLiteralIndex=0 |
| modeDeclaration SwitchInitiated | A point in time where the switch to the associated <code>ModeDeclarationGroupPrototype</code> has been initiated. Tags: atp.EnumerationLiteralIndex=1 |

Table 2.22: `TDEventModeDeclarationTypeEnum`

2.3.2.1.2.4 `TDEventTrigger`

[TPS_TIMEX_00039] `TDEventTrigger` specifies events observable at trigger ports

Upstream requirements: RS_TIMEX_00001, RS_TIMEX_00018

〔The element `TDEventTrigger` is used to specify events, namely the activation and release of triggers, observable at required and provided trigger ports.〕

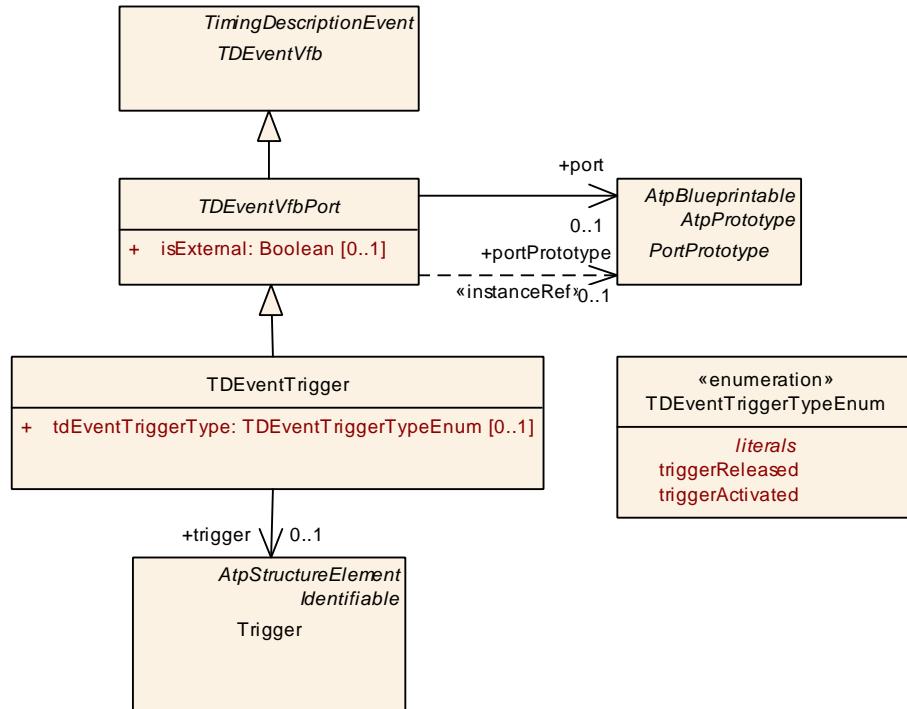


Figure 2.26: Trigger

| | | | | |
|----------------------|---|--------------|-------------|--|
| Class | TDEventTrigger | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescriptionEvents::TDEventVfb | | | |
| Note | A TimingDescriptionEvent triggered by a Trigger in a TriggerInterface on VFB level. | | | |
| Base | ARObject , Identifiable , MultilanguageReferrable , Referrable , TDEventVfb , TDEventVfbPort , TimingDescription , TimingDescriptionEvent | | | |
| Aggregated by | TimingExtension.timingDescription | | | |
| Attribute | Type | Mult. | Kind | Note |
| tdEventTriggerType | TDEventTriggerTypeEnum | 0..1 | attr | The specific type of this timing event. |
| trigger | Trigger | 0..1 | ref | The referenced Trigger from a TriggerInterface . |

Table 2.23: TDEventTrigger

[constr_6893] Existence of [TDEventTrigger.tdEventTriggerType](#)

Imposition time: [IT_VfbTd](#)

〔For each [TDEventTrigger](#), the attribute [tdEventTriggerType](#) shall exist〕

[constr_6894] Existence of [TDEventTrigger.trigger](#)

Imposition time: [IT_VfbTd](#)

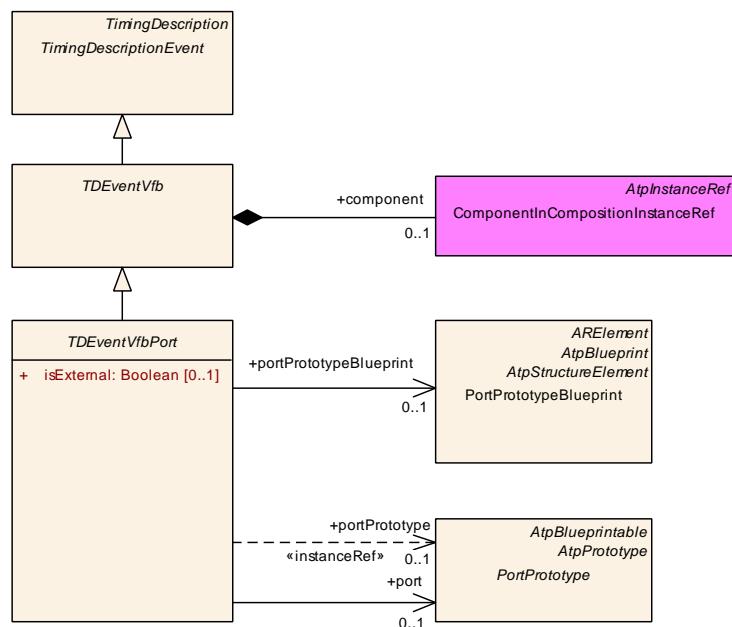
〔For each [TDEventTrigger](#), the reference to [Trigger](#) in the role [trigger](#) shall exist〕

| | |
|----------------------|--|
| Enumeration | TDEventTriggerTypeEnum |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescriptionEvents::TDEventVfb |
| Note | This is used to describe the specific event type of a TDEventTrigger. |
| Aggregated by | TDEventTrigger.tDEventTriggerType |
| Literal | Description |
| triggerActivated | A point in time where the referenced trigger has been successfully released and is activating runnable entities of the receiving SW-C. Tags: atp.EnumerationLiteralIndex=0 |
| triggerReleased | A point in time where the referenced trigger has been successfully released by the emitting SW-C. Tags: atp.EnumerationLiteralIndex=1 |

 Table 2.24: **TDEventTriggerTypeEnum**

2.3.2.1.2.5 Blueprinting TDEventVfbPort

The primary purpose of blueprinting [VfbTiming](#) is for use with a "AUTOSAR Application Interface" [7]. In a blueprint of an "AUTOSAR Application Interface", a [Timing-Constraint](#) can be applied to [TDEventVfb](#) to annotate a high-level timing goal for an blueprinted [PortPrototype](#). The reference [portPrototypeBlueprint](#) refers to a [PortPrototypeBlueprint](#) which effectively wraps a [PortPrototype](#) used in the context of an AUTOSAR Blueprint, but, permits certain extra aspects over a [PortPrototype](#) as explained in "Blueprinting PortPrototype" [4].


 Figure 2.27: **TDEventVfb Blueprint**

[constr_4508] Existence of `TDEventVfbPort.portPrototypeBlueprint`

Imposition time: `IT_VfbTd`

〔The reference `TDEventVfbPort.portPrototypeBlueprint` shall exist only if the immediate parent is `ARPackage.category==BLUEPRINT`〕

[constr_6900] Dual existence of `TDEventVfb.port` and `TDEventVfb.portPrototypeBlueprint`

Status: DRAFT

Imposition time: `IT_VfbTd`

〔The reference `TDEventVfbPort.port` and `TDEventVfbPort.portPrototypeBlueprint` shall not co-exist in a model〕

2.3.2.2 `TDEventSwc`

[TPS_TIMEX_00044] Purpose of `TDEventSwc`

Upstream requirements: `RS_TIMEX_00001`, `RS_TIMEX_00019`, `RS_TIMEX_00020`

〔The element `TDEventSwc` is used to specify events, namely the activation, start, termination of runnable entities, as well as variable accesses, which are observable in the Software Component view.〕

| | | | | |
|----------------------|---|--------------|-------------|--|
| Class | <code>TDEventSwc</code> (abstract) | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescriptionEvents::TDEventSwcInternalBehavior | | | |
| Note | This is the abstract parent class to describe timing events at Software Component (SW-C) level. | | | |
| Base | <code>ARObject</code> , <code>Identifiable</code> , <code>MultilanguageReferrable</code> , <code>Referrable</code> , <code>TimingDescription</code> , <code>TimingDescriptionEvent</code> | | | |
| Subclasses | <code>TDEventSwcInternalBehavior</code> , <code>TDEventSwcInternalBehaviorReference</code> | | | |
| Aggregated by | <code>TimingExtension.timingDescription</code> | | | |
| Attribute | Type | Mult. | Kind | Note |
| component | <code>SwComponentPrototype</code> | 0..1 | iref | The context for the scope of this timing event. InstanceRef implemented by: <code>ComponentInCompositionInstanceRef</code> |

Table 2.25: `TDEventSwc`

2.3.2.2.1 **TDEventSwcInternalBehavior**

[TPS_TIMEX_00020] **TDEventSwcInternalBehavior specifies observable events of runnable entities**

Upstream requirements: RS_TIMEX_00001, RS_TIMEX_00019, RS_TIMEX_00020

[The element **TDEventSwcInternalBehavior** is used to specify events, namely the activation, start, termination of runnable entities, as well as variable accesses, which are observable in the Software Component view.]

Events related to **SwcInternalBehavior** can be used during the specification of:

- [SwcTiming 2.1.2](#)
- [SystemTiming 2.1.3](#)
- [EcuTiming 2.1.5](#)

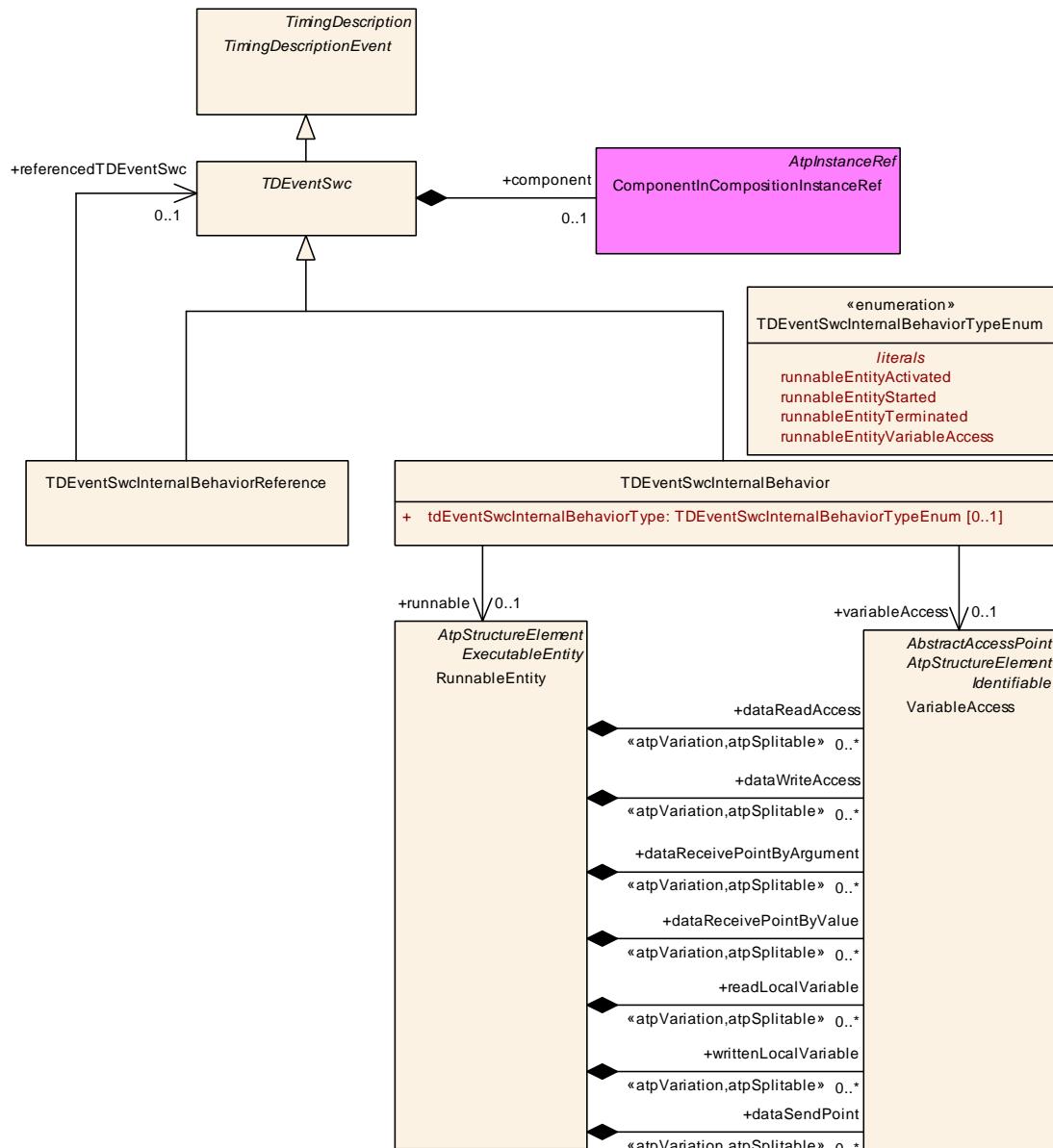


Figure 2.28: Event of type "SwInternalBehavior"

| Class | TDEventSwInternalBehavior | | | |
|---------------|---|-------|------|---------------------------------|
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescriptionEvents::TDEventSwInternalBehavior | | | |
| Note | This is used to describe timing events related to the SwInternalBehavior of an AtomicSwComponent Type. | | | |
| Base | <i>ARObject, Identifiable, MultilanguageReferrable, Referrable, TDEventSwc, TimingDescription, TimingDescriptionEvent</i> | | | |
| Aggregated by | <i>TimingExtension.timingDescription</i> | | | |
| Attribute | Type | Mult. | Kind | Note |
| runnable | RunnableEntity | 0..1 | ref | The scope of this timing event. |





| Class | TDEventSwcInternalBehavior | | | |
|--------------------------------|------------------------------------|------|------|---|
| tdEventSwcInternalBehaviorType | TDEventSwcInternalBehaviorTypeEnum | 0..1 | attr | The specific type of this timing event. |
| variableAccess | VariableAccess | 0..1 | ref | The scope of this timing event. |

Table 2.26: TDEventSwcInternalBehavior

[constr_6882] Existence of `TDEventSwcInternalBehavior.tdEventSwcInternalBehaviorType`

Imposition time: IT_SwcTd

〔For each `TDEventSwcInternalBehavior`, the attribute `tdEventSwcInternalBehaviorType` shall exist〕

[constr_6883] Existence of `TDEventSwcInternalBehavior.runnable`

Imposition time: IT_SwcTd

〔For each `TDEventSwcInternalBehavior`, the reference to `RunnableEntity` in the role `runnable` shall exist〕

| Enumeration | TDEventSwcInternalBehaviorTypeEnum |
|------------------------------|--|
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescriptionEvents::TDEventSwcInternalBehavior |
| Note | This is used to describe the specific event type of a TDEventSwcInternalBehavior. |
| Aggregated by | <code>TDEventSwcInternalBehavior.tdEventSwcInternalBehaviorType</code> |
| Literal | <i>Description</i> |
| runnableEntityActivated | A point in time where the associated RunnableEntity has been activated, which means that it has entered the state "to be started". Tags: atp.EnumerationLiteralIndex=0 |
| runnableEntityStarted | A point in time where the associated RunnableEntity has entered the state "started" after its activation. Tags: atp.EnumerationLiteralIndex=1 |
| runnableEntityTerminated | A point in time where the associated RunnableEntity has terminated and entered the state "suspended". Tags: atp.EnumerationLiteralIndex=2 |
| runnableEntityVariableAccess | A point in time where the associated variable is accessed. Tags: atp.EnumerationLiteralIndex=3 |

Table 2.27: TDEventSwcInternalBehaviorTypeEnum

[constr_4510] Specifying references to `RunnableEntity` and `VariableAccess`

Imposition time: IT_SwcTd

〔A `RunnableEntity` and `VariableAccess` shall be referenced at the same time if and only if the value of `tdEventSwcInternalBehaviorType==runnableEntityVariableAccess`. These two references are **not** mutual exclusive.〕

[constr_4511] Validity of referencing RunnableEntity

Imposition time: `IT_SwcTd`

「A `RunnableEntity` shall be referenced if and only if the value of `tdEventSwcInternalBehaviorType` is either:

- `runnableEntityActivated`
- `runnableEntityStarted`
- `runnableEntityTerminated`
- `runnableEntityVariableAccess`

」

[constr_4512] Validity of referencing VariableAccess

Imposition time: `IT_SwcTd`

「A `VariableAccess` shall be referenced if and only if the value of `tdEventSwcInternalBehaviorType==runnableEntityVariableAccess`.」

[TPS_TIMEX_00045] Purpose of TDEventSwcInternalBehaviorReference

Upstream requirements: `RS_TIMEX_00001, RS_TIMEX_00019`

「The element `TDEventSwcInternalBehaviorReference` is used to reference `TimingDescriptionEvent`s already specified in other timing views. In other words, it enables one to re-use existing timing models.」

| | | | | |
|-----------------------|--|--------------|-------------|--|
| Class | <code>TDEventSwcInternalBehaviorReference</code> | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescriptionEvents::TDEventSwcInternalBehavior | | | |
| Note | This is used to reference timing description events related to the Software Component (SW-C) view which are specified in other timing views. | | | |
| Base | <code>ARObject, Identifiable, MultilanguageReferrable, Referrable, TDEventSwc, TimingDescription, TimingDescriptionEvent</code> | | | |
| Aggregated by | <code>TimingExtension.timingDescription</code> | | | |
| Attribute | Type | Mult. | Kind | Note |
| referenced TDEventSwc | <code>TDEventSwc</code> | 0..1 | ref | The referenced timing description event. |

Table 2.28: TDEventSwcInternalBehaviorReference

[constr_6884] Existence of TDEventSwcInternalBehaviorReference.referencedTDEventSwc

Imposition time: `IT_SwcTd`

「For each `TDEventSwcInternalBehaviorReference`, the reference to `TDEventSwc` in the role `referencedTDEventSwc` shall exist」

2.3.2.3 TDEventCom

[TPS_TIMEX_00021] Purpose of [TDEventCom](#)

Upstream requirements: [RS_TIMEX_00001](#)

〔The element [TDEventCom](#) and its specializations are used to describe the occurrences of an event which are observed at a specific location in the System view, in particular any event related to communications.〕

Events related to communication can be used during the specification of:

- [SystemTiming 2.1.2](#)
- [EcuTiming 2.1.5](#)

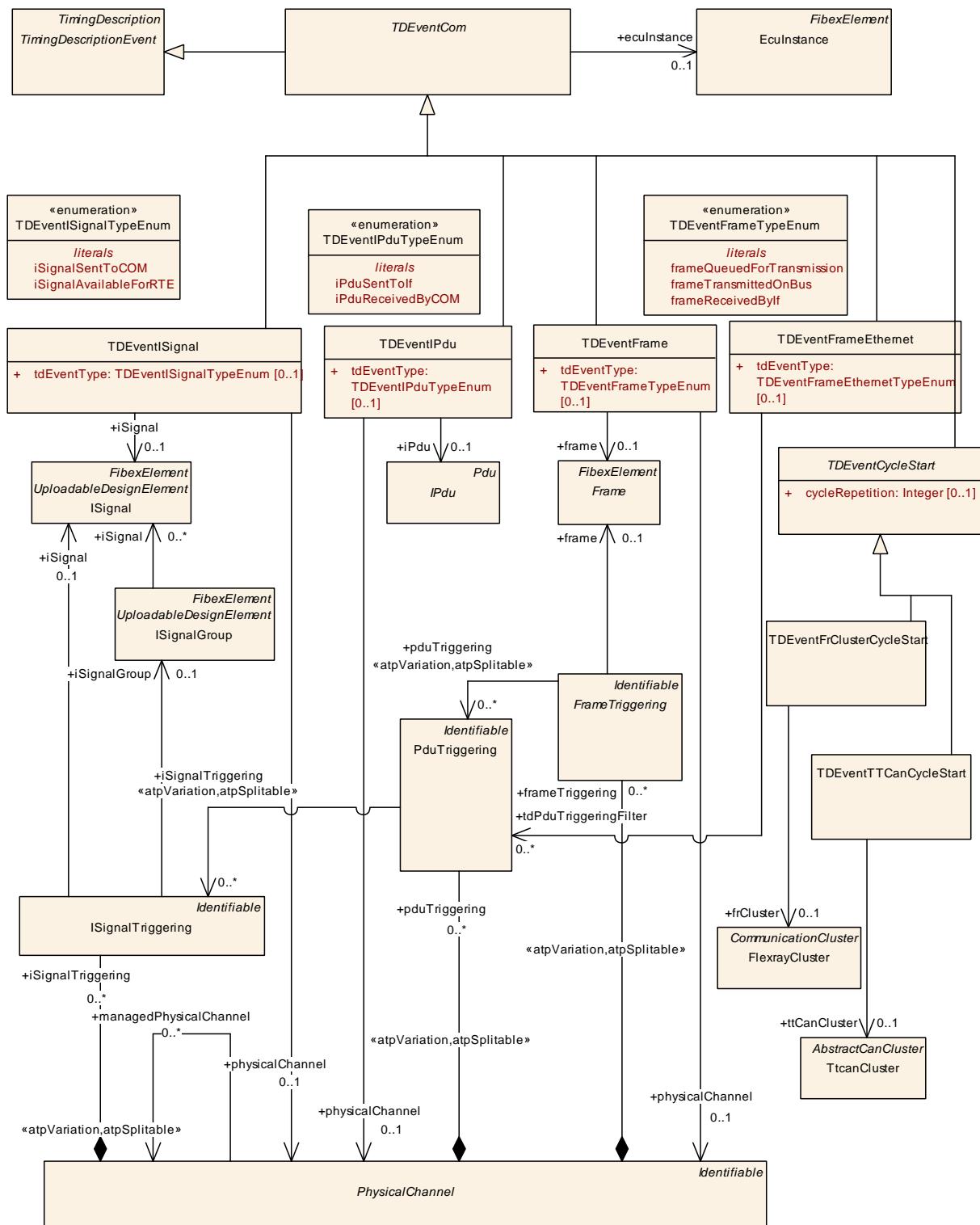


Figure 2.29: Events regarding communication

| | | | | |
|----------------------|---|--------------|-------------|---|
| Class | <i>TDEventCom</i> (abstract) | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescriptionEvents::TDEventCom | | | |
| Note | This is the abstract parent class to describe timing events related to communication including the physical layer. | | | |
| Base | ARObject, <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i> , <i>TimingDescription</i> , <i>TimingDescriptionEvent</i> | | | |
| Subclasses | <i>TDEventCycleStart</i> , <i>TDEventFrame</i> , <i>TDEventFrameEthernet</i> , <i>TDEventIPdu</i> , <i>TDEventISignal</i> | | | |
| Aggregated by | <i>TimingExtension.timingDescription</i> | | | |
| Attribute | Type | Mult. | Kind | Note |
| eculnstance | Eculnstance | 0..1 | ref | The ECU context for a particular timing event. The link is optional, because the Eculnstance can not be defined for events of type <i>TDEventCycleStart</i> . |

 Table 2.29: **TDEventCom**

2.3.2.3.1 TDEventISignal

[TPS_TIMEX_00022] *TDEventISignal* specifies events related to the exchange of *I-Signals*

Upstream requirements: [RS_TIMEX_00001](#)

〔The element *TDEventISignal* is used to specify events, namely the exchange of *ISignal*, observable between the RTE and the COM BSW module.〕

| | | | | |
|----------------------|---|--------------|-------------|---|
| Class | <i>TDEventISignal</i> | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescriptionEvents::TDEventCom | | | |
| Note | Describe timing events related to the exchange of <i>TDEventISignal</i> s between COM and RTE. | | | |
| Base | ARObject, <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i> , <i>TDEventCom</i> , <i>TimingDescription</i> , <i>TimingDescriptionEvent</i> | | | |
| Aggregated by | <i>TimingExtension.timingDescription</i> | | | |
| Attribute | Type | Mult. | Kind | Note |
| iSignal | <i>ISignal</i> | 0..1 | ref | The scope of this timing event. |
| physical Channel | <i>PhysicalChannel</i> | 0..1 | ref | The PhysicalChannel on which the <i>ISignal</i> is transmitted. |
| tdEventType | <i>TDEventISignalType</i> Enum | 0..1 | attr | The specific type of this timing event. |

 Table 2.30: **TDEventISignal**

[constr_6915] Affinity of *ISignal* in *TDEventISignal*

Status: DRAFT

Imposition time: [IT_EcuTd](#)

〔The referenced *ISignal* in the role *TDEventISignal.iSignal* shall exist also in the list of *iSignal*s aggregated by *TDEventISignal.physicalChannel.iSignalTriggering*〕

[constr_6864] Existence of `TDEventISignal.tdEventType`

Imposition time: `IT_EcuTd`

〔For each `TDEventISignal`, the attribute `tdEventType` shall exist〕

[constr_6865] Existence of `TDEventISignal.iSignal`

Imposition time: `IT_EcuTd`

〔For each `TDEventISignal`, the reference to `ISignal` in the role `iSignal` shall exist〕

[constr_6866] Existence of `TDEventISignal.physicalChannel`

Imposition time: `IT_EcuTd`

〔For each `TDEventISignal`, the reference to `PhysicalChannel` in the role `physicalChannel` shall exist〕

| <i>Enumeration</i> | <code>TDEventISignalTypeEnum</code> |
|-------------------------------------|--|
| <i>Package</i> | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescriptionEvents::TDEventCom |
| <i>Note</i> | This is used to describe the specific event type of a <code>TDEventISignal</code> . |
| <i>Aggregated by</i> | <code>TDEventISignal.tdEventType</code> |
| <i>Literal</i> | <i>Description</i> |
| <code>iSignalAvailableForRTE</code> | A point in time, where the COM module makes the contained signal / signal group available for the RTE and the corresponding Rx Indication callout is generated (if configured). Tags: <code>atp:EnumerationLiteralIndex=0</code> |
| <code>iSignalSentToCOM</code> | A point in time, where a transmission request call is issued by the RTE on a named COM signal / signal group and the new value is stored to the carrier COM I-PDU buffer. Tags: <code>atp:EnumerationLiteralIndex=1</code> |

Table 2.31: `TDEventISignalTypeEnum`

2.3.2.3.2 `TDEventIPdu`

[TPS_TIMEX_00023] `TDEventIPdu` specifies events related to the exchange of I-PDUs

Upstream requirements: `RS_TIMEX_00001`

〔The element `TDEventIPdu` is used to specify events, namely the exchange of I-PDUs, observable between the bus specific BSW modules (CAN, FlexRay, LIN) and the AUTOSAR Com.〕

| | | | | |
|----------------------|--|--------------|-------------|---|
| Class | TDEventIPdu | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescription Events::TDEventCom | | | |
| Note | Describe timing events related to the exchange of IPdu s between the bus specific (FlexRay / CAN / LIN) Interface BSW module and COM. | | | |
| Base | ARObject , Identifiable , MultilanguageReferrable , Referrable , TDEventCom , TimingDescription , TimingDescriptionEvent | | | |
| Aggregated by | TimingExtension.timingDescription | | | |
| Attribute | Type | Mult. | Kind | Note |
| iPdu | IPdu | 0..1 | ref | The scope of this timing event. |
| physical Channel | PhysicalChannel | 0..1 | ref | The PhysicalChannel on which the IPdu is transmitted. |
| tdEventType | TDEventIPduTypeEnum | 0..1 | attr | The specific type of this timing event. |

Table 2.32: TDEventIPdu
[constr_6917] Affinity of [IPdu](#) in [TDEventIPdu](#)
Status: DRAFT

Imposition time: [IT_EcuTd](#)

〔The referenced [IPdu](#) in the role [TDEventIPdu.iPdu](#) shall exist also in the list of [IPdu](#)s aggregated by [TDEventIPdu.physicalChannel.pduTriggering](#)〕

[constr_6867] Existence of [TDEventIPdu.tdEventType](#)
Imposition time: [IT_EcuTd](#)

〔For each [TDEventIPdu](#), the attribute [tdEventType](#) shall exist〕

[constr_6868] Existence of [TDEventIPdu.iPdu](#)
Imposition time: [IT_EcuTd](#)

〔For each [TDEventIPdu](#), the reference to [IPdu](#) in the role [iPdu](#) shall exist〕

[constr_6869] Existence of [TDEventIPdu.physicalChannel](#)
Imposition time: [IT_EcuTd](#)

〔For each [TDEventIPdu](#), the reference to [PhysicalChannel](#) in the role [physicalChannel](#) shall exist〕

| | |
|----------------------|--|
| Enumeration | TDEventIPduTypeEnum |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescription Events::TDEventCom |
| Note | This is used to describe the specific event type of a TDEventIPdu. |
| Aggregated by | TDEventIPdu.tdEventType |
| Literal | Description |





| Enumeration | TDEventIPduTypeEnum |
|--------------------|---|
| iPduReceivedBy COM | A point in time where the received frame is processed by the corresponding (FlexRay / CAN / LIN) Interface BSW module, routed through the PDUR and the contained PDUs are pushed to the COM module. Tags: atp:EnumerationLiteralIndex=0 |
| iPduSentTolff | A point in time where the carrier COM I-PDU is routed through the PDUR and is pushed to the bus specific (FlexRay / CAN / LIN) Interface BSW module. Tags: atp:EnumerationLiteralIndex=1 |

Table 2.33: TDEventIPduTypeEnum

2.3.2.3.3 TDEventFrame

[TPS_TIMEX_00024] [TDEventFrame](#) specifies events related to the exchange of network frames

Upstream requirements: [RS_TIMEX_00001](#)

[The element [TDEventFrame](#) is used to specify events, namely the exchange of Frames, observable between the communication controller and the bus specific BSW modules (CAN, FlexRay, LIN) and observable at the physical layer.]

| Class | TDEventFrame | | | |
|------------------|---|-------|------|--|
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescriptionEvents::TDEventCom | | | |
| Note | This is used to describe timing events related to the exchange of frames between the communication controller and the bus specific (FlexRay / CAN / LIN) Interface BSW module. | | | |
| Base | ARObject, Identifiable , MultilanguageReferrable , Referrable , TDEventCom , TimingDescription , TimingDescriptionEvent | | | |
| Aggregated by | TimingExtension.timingDescription | | | |
| Attribute | Type | Mult. | Kind | Note |
| frame | Frame | 0..1 | ref | The scope of this timing event. |
| physical Channel | PhysicalChannel | 0..1 | ref | The PhysicalChannel on which the Frame is transmitted. |
| tdEventType | TDEventFrameType Enum | 0..1 | attr | The specific type of this timing event. |

Table 2.34: TDEventFrame

[constr_6916] Affinity of Frame in [TDEventFrame](#)

Status: DRAFT

Imposition time: [IT_EcuTd](#)

[The referenced [Frame](#) in the role [TDEventFrame.frame](#) shall exist also in the list of [frame](#)s aggregated by [TDEventFrame.physicalChannel.frameTriggering](#)]

[constr_6870] Existence of `TDEventFrame.tdEventType`

Imposition time: `IT_EcuTd`

For each `TDEventFrame`, the attribute `tdEventType` shall exist]

[constr_6871] Existence of `TDEventFrame.frame`

Imposition time: `IT_EcuTd`

For each `TDEventFrame`, the reference to `Frame` in the role `frame` shall exist]

[constr_6872] Existence of `TDEventFrame.physicalChannel`

Imposition time: `IT_EcuTd`

For each `TDEventFrame`, the reference to `PhysicalChannel` in the role `physicalChannel` shall exist]

| <i>Enumeration</i> | <code>TDEventFrameTypeEnum</code> |
|----------------------------|---|
| <i>Package</i> | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescriptionEvents::TDEventCom |
| <i>Note</i> | This is used to describe the specific event type of a <code>TDEventFrame</code> . |
| <i>Aggregated by</i> | <code>TDEventFrame.tdEventType</code> |
| <i>Literal</i> | <i>Description</i> |
| frameQueuedForTransmission | A point in time where the frame containing the named signal / I-PDU is queued for transmission within the related Communication Driver. Tags: <code>atp.EnumerationLiteralIndex=0</code> |
| frameReceivedByIf | A point in time where the frame is pushed from the subscriber's communication controller to the corresponding (FlexRay / CAN / LIN) Interface BSW module. Tags: <code>atp.EnumerationLiteralIndex=1</code> |
| frameTransmittedOnBus | A point in time where the transmission of the frame completes successfully, and the subscriber's communication controller receives the frame from the bus. Tags: <code>atp.EnumerationLiteralIndex=2</code> |

Table 2.35: `TDEventFrameTypeEnum`

2.3.2.3.4 TDEventComEthernet

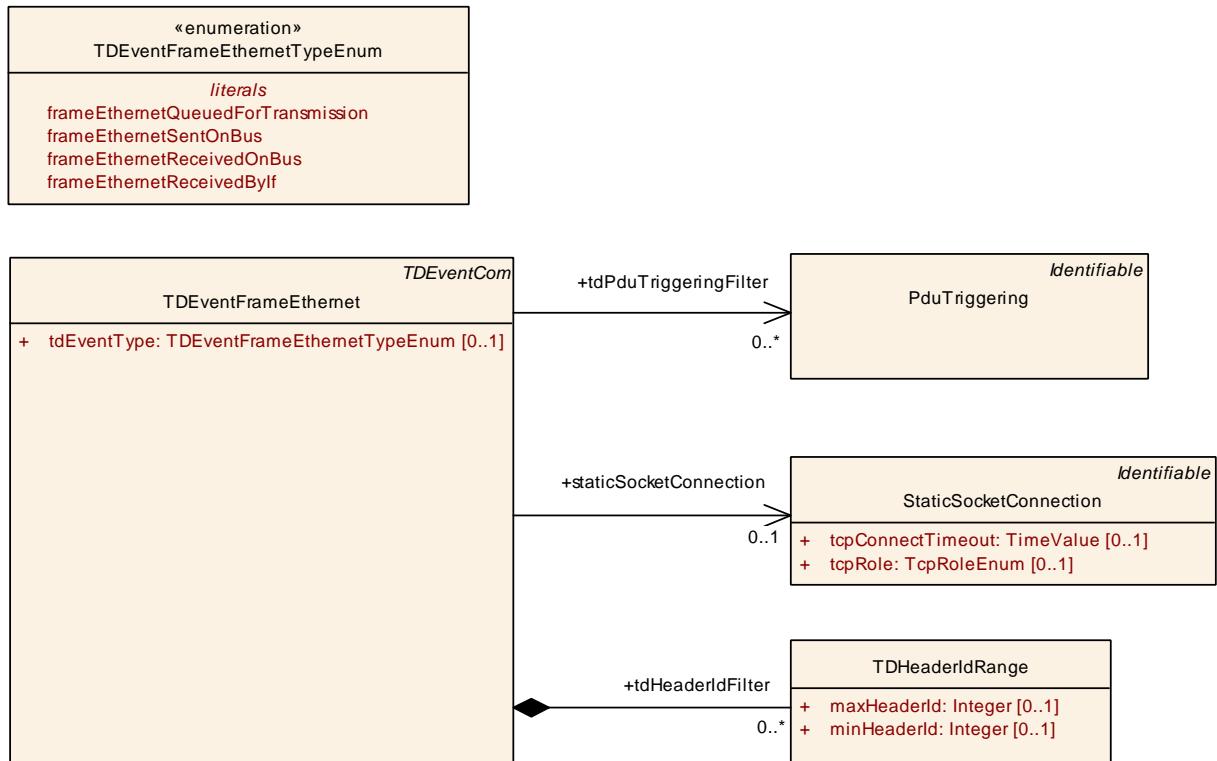


Figure 2.30: Events regarding Ethernet communication

[TPS_TIMEX_00052] **TDEventFrameEthernet** specifies events related to the exchange of Ethernet frames

Upstream requirements: [RS_TIMEX_00001](#)

The element **TDEventFrameEthernet** is used to specify events, namely the exchange of Ethernet frames, observable between the Ethernet communication controller and the Ethernet specific BSW modules, as well as observable at the physical layer.]

| Class | TDEventFrameEthernet | | | |
|------------------------|---|-------|------|--|
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescriptionEvents::TDEventCom | | | |
| Note | This is used to describe timing description events related to the exchange of Ethernet frames between an Ethernet communication controller and the BSW Ethernet interface and driver module. | | | |
| Base | ARObject, Identifiable , MultilanguageReferrable , Referrable , TDEventCom , TimingDescription , TimingDescriptionEvent | | | |
| Aggregated by | TimingExtension.timingDescription | | | |
| Attribute | Type | Mult. | Kind | Note |
| staticSocketConnection | StaticSocketConnection | 0..1 | ref | Specifies the SocketConnection by the means of which Physical Data Units (PDU) are transmitted or received within an Ethernet Frame. |
| tdEventType | TDEventFrameEthernetTypeEnum | 0..1 | attr | This is used to describe the specific event type of a TDEventFrameEthernet. |





| Class | TDEventFrameEthernet | | | |
|------------------------|----------------------|---|------|--|
| tdHeaderIdFilter | TDHeaderIdRange | * | aggr | Specifies the header identifier or a range of header identifiers that if contained in the Ethernet frame let the TDEventFrameEthernet occur. |
| tdPduTriggering Filter | PduTriggering | * | ref | Specifies the PDU that if contained in the Ethernet frame let the TDEventFrameEthernet occur. |

Table 2.36: TDEventFrameEthernet
[constr_6873] Existence of TDEventFrameEthernet.tdEventType
Imposition time: IT_EcuTd

For each TDEventFrameEthernet, the attribute `tdEventType` shall exist]

| Enumeration | TDEventFrameEthernetTypeEnum |
|--------------------------------------|--|
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescription Events::TDEventCom |
| Note | This is used to describe the specific event type of a TDEventFrameEthernet. |
| Aggregated by | TDEventFrameEthernet.tdEventType |
| Literal | Description |
| frameEthernet QueuedFor Transmission | A point in time where the Ethernet frame containing the specified PDUs is queued for transmission within the corresponding Ethernet Communication Driver. Tags: atp.EnumerationLiteralIndex=0 |
| frameEthernet ReceivedByIf | A point in time where the frame is pushed from the corresponding Ethernet communication controller to the BSW Ethernet communication interface. Tags: atp.EnumerationLiteralIndex=1 |
| frameEthernet ReceivedOnBus | A point in time where the receipt of the Ethernet frame/packet completes successfully on the recipient's Ethernet communication controller. In other words, the Ethernet frame/packet has entered the recipient's Ethernet communication controller which means the last bit of the Ethernet frame/packet has been received. Tags: atp.EnumerationLiteralIndex=2 |
| frameEthernetSent OnBus | A point in time where the transmission of the Ethernet frame/packet completes successfully on the physical Ethernet communication network. In other words, the Ethernet frame/packet has left the sender's Ethernet communication controller, which means that the last bit of the Ethernet frame/packet has been sent. Tags: atp.EnumerationLiteralIndex=3 |

Table 2.37: TDEventFrameEthernetTypeEnum

| Class | TDHeaderIdRange | | | |
|---------------|--|-------|------|------|
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescription Events::TDEventCom | | | |
| Note | Specifies a range of PDU header identifiers. This range is specified by a minimum and maximum header identifier; and the maximum header identifier shall be greater than or equal the minimum header identifier. | | | |
| Base | ARObject | | | |
| Aggregated by | TDEventFrameEthernet.tdHeaderIdFilter | | | |
| Attribute | Type | Mult. | Kind | Note |





| Class | TDHeaderIdRange | | | |
|-------------|-----------------|------|------|---|
| maxHeaderId | Integer | 0..1 | attr | Specifies the maximum PDU header identifier, in other words the upper bound of a range of PDU header identifiers. |
| minHeaderId | Integer | 0..1 | attr | Specifies the minimum PDU header identifier, in other words the lower bound of a range of PDU header identifiers. |

Table 2.38: TDHeaderIdRange

[constr_6874] Existence of TDHeaderIdRange.maxHeaderId

Imposition time: [IT_EcuTd](#)

For each TDHeaderIdRange, the attribute [maxHeaderId](#) shall exist]

[constr_6875] Existence of TDHeaderIdRange.minHeaderId

Imposition time: [IT_EcuTd](#)

For each TDHeaderIdRange, the attribute [minHeaderId](#) shall exist]

2.3.2.3.5 TDEventCycleStart

[constr_6876] Existence of TDEventCycleStart.cycleRepetition

Imposition time: [IT_EcuTd](#)

For each TDEventCycleStart, the attribute [cycleRepetition](#) shall exist]

| | | | | |
|----------------------|--|--------------|-------------|---|
| Class | TDEventCycleStart (abstract) | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescriptionEvents::TDEventCom | | | |
| Note | <p>This is the abstract parent class to describe timing events related to a point in time where a communication cycle starts.</p> <p>Via the attribute "cycleRepetition", a filtered view to the cycle start can be defined.</p> | | | |
| Base | ARObject , Identifiable , MultilanguageReferrable , Referrable , TDEventCom , TimingDescription , TimingDescriptionEvent | | | |
| Subclasses | TDEventFrClusterCycleStart , TDEventTTCanCycleStart | | | |
| Aggregated by | TimingExtension.timingDescription | | | |
| Attribute | Type | Mult. | Kind | Note |
| cycleRepetition | Integer | 0..1 | attr | The start of every <cycleRepetition> cycle is targeted by this event. |

Table 2.39: TDEventCycleStart

2.3.2.3.5.1 TDEventFrClusterCycleStart

[TPS_TIMEX_00025] `TDEventFrClusterCycleStart` specifies the event related to the start of a FlexRay communication cycle

Upstream requirements: RS_TIMEX_00001

〔The element `TDEventFrClusterCycleStart` is used to specify events, namely the start of a communication cycle, observable at the physical layer of the FlexRay bus.〕

| | | | | |
|----------------------|--|--------------|-------------|---------------------------------|
| Class | <code>TDEventFrClusterCycleStart</code> | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescriptionEvents::TDEventCom | | | |
| Note | This is used to describe the timing event related to a point in time where a communication cycle starts on a FlexRay cluster. | | | |
| Base | <i>ARObject, Identifiable, MultilanguageReferrable, Referrable, TDEventCom, TDEventCycleStart, TimingDescription, TimingDescriptionEvent</i> | | | |
| Aggregated by | <i>TimingExtension.timingDescription</i> | | | |
| Attribute | Type | Mult. | Kind | Note |
| frCluster | FlexrayCluster | 0..1 | ref | The scope of this timing event. |

Table 2.40: `TDEventFrClusterCycleStart`

[constr_6877] Existence of `TDEventFrClusterCycleStart.frCluster`

Imposition time: IT_EcuTd

〔For each `TDEventFrClusterCycleStart`, the attribute `frCluster` shall exist〕

2.3.2.3.5.2 TDEventTTCanCycleStart

[TPS_TIMEX_00026] `TDEventTTCanCycleStart` specifies the event related to the start of a TTCAN communication cycle

Upstream requirements: RS_TIMEX_00001

〔The element `TDEventTTCanCycleStart` is used to specify events, namely the start of a communication cycle, observable at the physical layer of the TTCAN bus.〕

| | | | | |
|----------------|---|--|--|--|
| Class | <code>TDEventTTCanCycleStart</code> | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescriptionEvents::TDEventCom | | | |
| Note | This is used to describe the timing event related to a point in time where a communication cycle starts on a TTCAN cluster. | | | |



△

| Class | TDEventTTCanCycleStart | | | |
|---------------|---|-------|------|---------------------------------|
| Base | ARObject, Identifiable, MultilanguageReferrable, Referrable, TDEventCom, TDEventCycleStart, TimingDescription, TimingDescriptionEvent | | | |
| Aggregated by | TimingExtension.timingDescription | | | |
| Attribute | Type | Mult. | Kind | Note |
| ttCanCluster | TtcanCluster | 0..1 | ref | The scope of this timing event. |

Table 2.41: TDEventTTCanCycleStart

[constr_6878] Existence of `TDEventTTCanCycleStart.ttCanCluster`
Imposition time: `IT_EcuTd`

〔For each `TDEventTTCanCycleStart`, the attribute `ttCanCluster` shall exist〕

2.3.2.4 TDEventBswInternalBehavior

[TPS_TIMEX_00028] `TDEventBswInternalBehavior` specifies observable events of BSW module entities
Upstream requirements: `RS_TIMEX_00001`

〔The element `TDEventBswInternalBehavior` is used to specify events, namely the activation, start and termination of BSW module entities, which are observable in the Basic Software Module view.〕

Events related to the BSW can be used during the specification of:

- `BswModuleTiming` 2.1.4.1
- `EcuTiming` 2.1.5

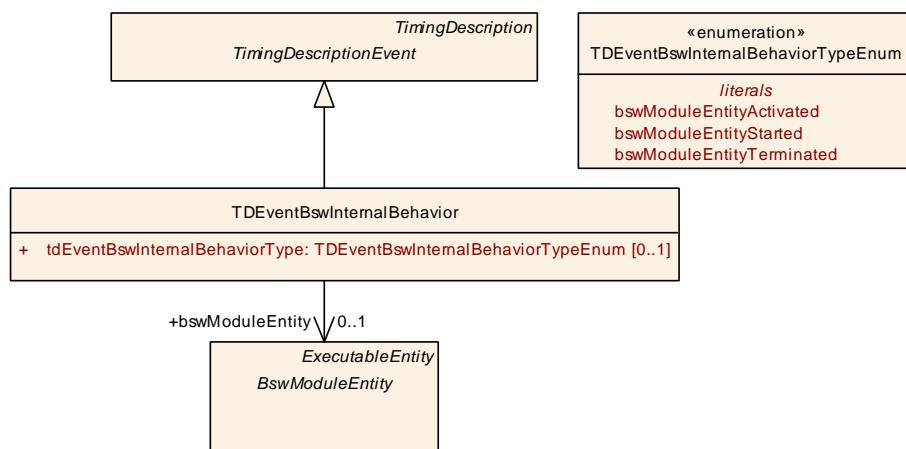


Figure 2.31: Events related to the internal structure of a BSW module

| | | | | |
|--------------------------------|---|--------------|-------------|---|
| Class | TDEventBswInternalBehavior | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescriptionEvents::TDEventBswInternalBehavior | | | |
| Note | This is used to describe timing events related to the BswInternalBehavior of a BSW module. | | | |
| Base | ARObject, Identifiable, MultilanguageReferrable, Referrable, TimingDescription, TimingDescriptionEvent | | | |
| Aggregated by | TimingExtension.timingDescription | | | |
| Attribute | Type | Mult. | Kind | Note |
| bswModuleEntity | BswModuleEntity | 0..1 | ref | The scope of this timing event. |
| tdEventBswInternalBehaviorType | TDEventBswInternalBehaviorTypeEnum | 0..1 | attr | The specific type of this timing event. |

Table 2.42: TDEventBswInternalBehavior

[constr_6858] Existence of TDEventBswInternalBehavior.tdEventBswInternalBehaviorType

Imposition time: IT_BswTd

[For each TDEventBswInternalBehavior, the attribute tdEventBswInternalBehaviorType shall exist]

[constr_6859] Existence of TDEventBswInternalBehavior.bswModuleEntity

Imposition time: IT_BswTd

[For each TDEventBswInternalBehavior, the reference to BswModuleEntity in the role bswModuleEntity shall exist]

Please note: For every TDEventBswInternalBehavior its scope is defined by the bswModuleEntity reference. It points to the BSW module entity for which the event can be observed. This scope definition assumes that every BSW module exists only once on each ECU. Otherwise the scope would not be precise enough because every module instance would bring the same BSW module entities.

| | | | | |
|---------------------------|---|--|--|--|
| Enumeration | TDEventBswInternalBehaviorTypeEnum | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescriptionEvents::TDEventBswInternalBehavior | | | |
| Note | This is used to describe the specific event type of a TDEventBswInternalBehavior. | | | |
| Aggregated by | TDEventBswInternalBehavior.tdEventBswInternalBehaviorType | | | |
| Literal | Description | | | |
| bswModuleEntity Activated | A point in time where the associated BswModuleEntity has been activated, which means that it has entered the state "to be started". Tags: atp.EnumerationLiteralIndex=0 | | | |
| bswModuleEntity Started | A point in time where the associated BswModuleEntity has entered the state "started" after its activation. Tags: atp.EnumerationLiteralIndex=1 | | | |





| Enumeration | TDEventBswInternalBehaviorTypeEnum |
|----------------------------|---|
| bswModuleEntity Terminated | A point in time where the associated BswModuleEntity has terminated and entered the state "suspended" Tags: atp.EnumerationLiteralIndex=2 |

Table 2.43: TDEventBswInternalBehaviorTypeEnum

2.3.2.5 TDEventBsw

[TPS_TIMEX_00029] Purpose of TDEventBsw

Upstream requirements: [RS_TIMEX_00001](#)

〔The element [TDEventBsw](#) is used to specify events which are observable in the Basic Software Module view, which means that the occurrences of such events are observable between the Basic Software Modules.〕

[constr_6901] Existence of TDEventBsw.bswModuleDescription

Status: DRAFT

Imposition time: [IT_BswTd](#)

〔For each [BswModuleTiming](#), the reference to a [BswModuleDescription](#) in the role [bswModuleDescription](#) shall exist〕

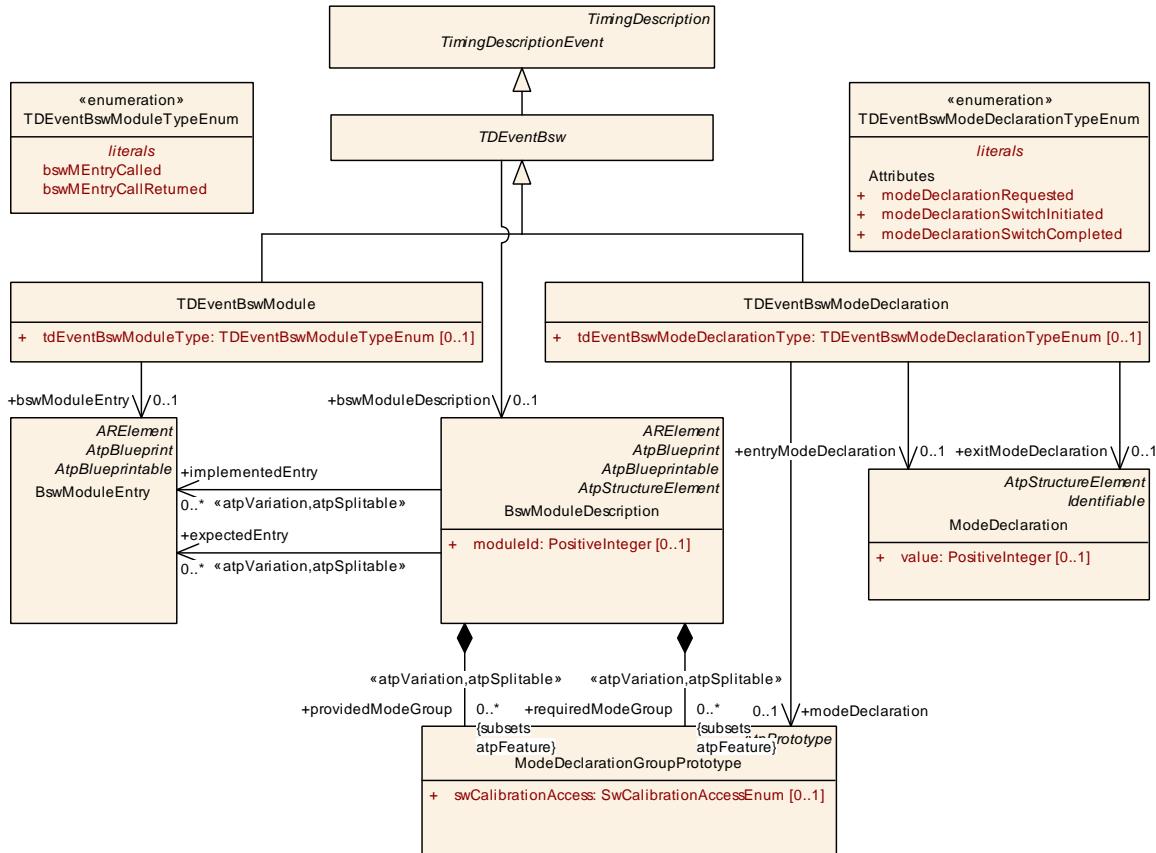


Figure 2.32: Events dealing with inter BSW module relations and mode communications on BSW level

2.3.2.5.1 TDEventBswModule

[TPS_TIMEX_00030] **TDEventBswModule** specifies observable events when basic software entries are called

Upstream requirements: [RS_TIMEX_00001](#)

The element **TDEventBswModule** is used to specify events, namely the calling of and return from called basic software module entries, observable when such entries are called within the Basic Software.]

| | |
|----------------------|--|
| Class | TDEventBswModule |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescriptionEvents::TDEventBsw |
| Note | This is used to describe timing events related to the interaction between BSW modules. |
| Base | <i>ARObject, Identifiable, MultilanguageReferable, Referrable, TDEventBsw, TimingDescription, TimingDescriptionEvent</i> |
| Aggregated by | <i>TimingExtension.timingDescription</i> |





| Class | TDEventBswModule | | | |
|----------------------|--------------------------|-------|------|---|
| Attribute | Type | Mult. | Kind | Note |
| bswModuleEntry | BswModuleEntry | 0..1 | ref | The scope of this timing event. |
| tdEventBswModuleType | TDEventBswModuleTypeEnum | 0..1 | attr | The specific type of this timing event. |

Table 2.44: TDEventBswModule

[constr_6860] Existence of TDEventBswModule.tdEventBswModuleType

Imposition time: IT_BswTd

For each TDEventBswModule, the attribute tdEventBswModuleType shall exist]

[constr_6861] Existence of TDEventBswModule.bswModuleEntry

Imposition time: IT_BswTd

For each TDEventBswModule, the reference to BswModuleEntry in the role bswModuleEntry shall exist]

| Enumeration | TDEventBswModuleTypeEnum |
|------------------------|---|
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescriptionEvents::TDEventBsw |
| Note | This is used to describe the specific event type of a TDEventBswModule. |
| Aggregated by | TDEventBswModule.tdEventBswModuleType |
| Literal | Description |
| bswMEntryCalled | A point in time where the associated BswModuleEntry has been called. Tags: atp.EnumerationLiteralIndex=0 |
| bswMEntryCall Returned | A point in time where the call of the associated BswModuleEntry has returned. Tags: atp.EnumerationLiteralIndex=1 |

Table 2.45: TDEventBswModuleTypeEnum

2.3.2.5.2 TDEventBswModeDeclaration

[TPS_TIMEX_00031] TDEventBswModeDeclaration specifies observable events in case of BSW mode communication

Upstream requirements: RS_TIMEX_00001

For the element TDEventBswModeDeclaration is used to specify events that are observable when mode changes are initiated and propagated in the Basic Software.]

| | | | | |
|---------------------------------|--|--------------|-------------|---|
| Class | TDEventBswModeDeclaration | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescriptionEvents::TDEventBsw | | | |
| Note | This is used to describe timing events related to the mode communication on BSW level. | | | |
| Base | ARObject, Identifiable, MultilanguageReferrable, Referrable, TDEventBsw, TimingDescription, TimingDescriptionEvent | | | |
| Aggregated by | <i>TimingExtension.timingDescription</i> | | | |
| Attribute | Type | Mult. | Kind | Note |
| entryMode Declaration | ModeDeclaration | 0..1 | ref | Optional parameter which refines the scope of the TDEventBswModeDeclaration. If the parameter is set, the event occurs only if the mode declaration group prototype instance shall enter into the referenced ModeDeclaration. |
| exitMode Declaration | ModeDeclaration | 0..1 | ref | Optional parameter which refines the scope of the TDEventBswModeDeclaration. If the parameter is set, the event occurs only if the mode declaration group prototype instance shall exit from the referenced ModeDeclaration. |
| mode Declaration | ModeDeclarationGroup Prototype | 0..1 | ref | The scope of this timing event. |
| tdEventBsw Mode DeclarationType | TDEventBswModeDeclarationTypeEnum | 0..1 | attr | The specific type of this timing event. |

Table 2.46: TDEventBswModeDeclaration

[constr_6862] Existence of `TDEventBswModeDeclaration.tDEventBswModeDeclarationType`

Imposition time: `IT_BswTd`

For each `TDEventBswModeDeclaration`, the attribute `tdEventBswModeDeclarationType` shall exist]

[constr_6863] Existence of `TDEventBswModeDeclaration.modeDeclaration`

Imposition time: `IT_BswTd`

For each `TDEventBswModeDeclaration`, the reference to `ModeDeclarationGroupPrototype` in the role `modeDeclaration` shall exist]

| | |
|---------------------------------|---|
| Enumeration | TDEventBswModeDeclarationTypeEnum |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescriptionEvents::TDEventBsw |
| Note | This is used to describe the specific event type of a TDEventBswModeDeclaration. |
| Aggregated by | <i>TDEventBswModeDeclaration.tDEventBswModeDeclarationType</i> |
| Literal | Description |
| modeDeclaration Requested | A point in time where the associated ModeDeclarationGroupPrototype has been requested. Tags: <code>atp.EnumerationLiteralIndex=0</code> |
| modeDeclaration SwitchCompleted | A point in time where the switch to the associated ModeDeclarationGroupPrototype has been completed. Tags: <code>atp.EnumerationLiteralIndex=1</code> |



| Enumeration | TDEventBswModeDeclarationTypeEnum |
|------------------------------------|---|
| modeDeclaration SwitchInitiated | A point in time where the switch to the associated ModeDeclarationGroupPrototype has been initiated by the BswM. Tags: atp.EnumerationLiteralIndex=2 |

Table 2.47: TDEventBswModeDeclarationTypeEnum

2.3.2.6 TDEventComplex

[TPS_TIMEX_00027] Purpose of **TDEventComplex**

Upstream requirements: RS_TIMEX_00001

〔The element **TDEventComplex** is used to specify relationships between occurrences of events.〕

Complex timing events can be used during the specification of:

- [VfbTiming 2.1.1](#)
- [SystemTiming 2.1.3](#)
- [SwcTiming 2.1.2](#)
- [BswModuleTiming 2.1.4.1](#)
- [EcuTiming 2.1.5](#)

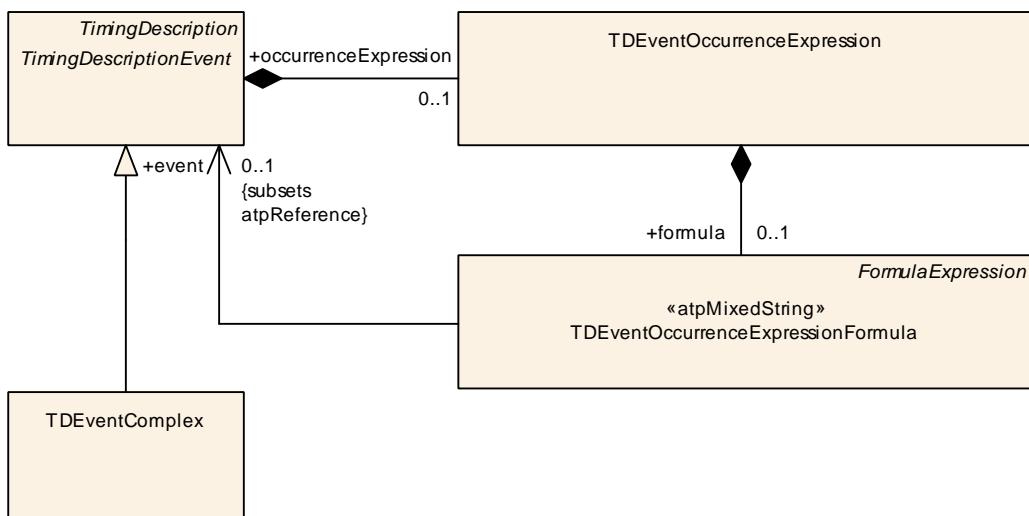


Figure 2.33: Complex timing event

| | | | | |
|----------------------|---|--------------|-------------|-------------|
| Class | TDEventComplex | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescriptionEvents::TDEventComplex | | | |
| Note | <p>This is used to describe complex timing events.</p> <p>The context of a complex timing event either is described informally, e.g. using the documentation block, or is described formally by the associated TDEventOccurrenceExpression.</p> | | | |
| Base | <i>ARObject, Identifiable, MultilanguageReferrable, Referrable, TimingDescription, TimingDescriptionEvent</i> | | | |
| Aggregated by | <i>TimingExtension.timingDescription</i> | | | |
| Attribute | Type | Mult. | Kind | Note |
| – | – | – | – | – |

Table 2.48: TDEventComplex

A complex timing event is a special observable event. In comparison to the "atomic" events described above a complex event does not contain information about the context it references, like [VariableDataPrototype](#) in [TDEventVariableDataPrototype](#). Instead, a complex event uses the occurrence expression to specify the context with regard to occurrences of [TimingDescriptionEvents](#) as described in the following section.

2.3.2.7 Occurrence Expression Language for Timing Events

The [TimingDescriptionEvents](#) mentioned in the previous sections allow to specify observable events with a well-defined context. However, sometimes the context information of the events is not sufficient, because additional conditions, like a value filter or additional stimuli, influence the occurrence. Thus, the occurrence expression provides means to overcome the limitations of atomic events.

The occurrence expression provides the ability to refine the context specification of a [TimingDescriptionEvent](#) for the following cases:

- [Content Filter](#): filters occurrences of an atomic event based on the *value* of exchanged data or operation arguments.
- [Complex Event](#): combines any number of atomic and complex events to specify a new [TimingDescriptionEvent](#).

2.3.2.7.1 Specifying an Occurrence Expression

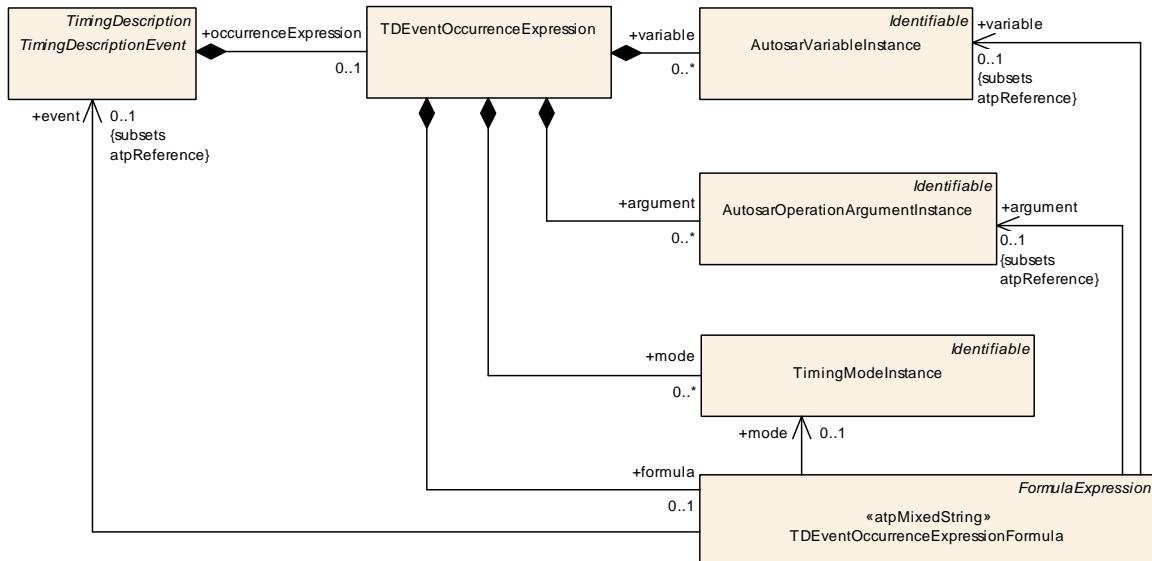


Figure 2.34: The occurrence expression

As shown in Figure 2.34, each `TimingDescriptionEvent` aggregates a `TDEventOccurrenceExpression` as an optional parameter. A `TDEventOccurrenceExpression` is a container for all information required to formulate the expression. The expression itself is defined via `TDEventOccurrenceExpressionFormula` which is derived from `FormulaExpression` (see Generic Structure Template [8]). The `TDEventOccurrenceExpressionFormula` uses the capabilities of the `FormulaExpression` and adds the following functions to the expression language:

- `TIMEX_value`:
 - operand:
 - * a `AutosarVariableInstance` or
 - * a `AutosarOperationArgumentInstance`
 - returns: `Numerical` (see [constr_4551]).
- `TIMEX_occurs`:
 - operand: a `TimingDescriptionEvent` whose occurrence shall be evaluated
 - returns: `Boolean`, TRUE if the referenced `TimingDescriptionEvent` occurs at the point in time the expression is evaluated, otherwise FALSE.
- `TIMEX_hasOccurred`:
 - operand: a `TimingDescriptionEvent` whose occurrence shall be evaluated

- returns: `Boolean`, TRUE if the referenced `TimingDescriptionEvent` has occurred *at least once* before or at the same point in time the expression is evaluated, otherwise FALSE.
- `TIMEX_timeSinceLastOccurrence`:
 - operand: a `TimingDescriptionEvent` whose occurrence shall be evaluated
 - returns: `Float` and the unit is seconds. It returns the time difference between the point in time of the last occurrence of the referenced event and the point in time the expression is evaluated.
- `TIMEX_angleSinceLastOccurrence`:
 - operand: a `TimingDescriptionEvent` whose occurrence shall be evaluated
 - returns: `Float` and the unit is degree. It returns the angle of the crank shaft between the point in time of the last occurrence of the referenced event and the point in time the expression is evaluated.
- `TIMEX_modeActive`:
 - operand: a `TimingModeInstance` whose occurrence shall be evaluated
 - returns: `Boolean`, TRUE if the specified mode declaration is *active* at the point in time the expression is evaluated, otherwise FALSE.

The starting point of the time interval considered by the `TIMEX` functions is the point in time the measurement of the event occurrences has been started.

All operands required by the functions are references to model elements. Thus, `TDEventOccurrenceExpressionFormula` requires references to the respective elements of type `TimingDescriptionEvent`, `AutosarVariableInstance`, `AutosarOperationArgumentInstance`, and `TimingModeInstance`. Due to the `atpMixedString` nature of the `TDEventOccurrenceExpressionFormula` several references can be used within the occurrence expression.

[constr_4500] Restricted usage of Occurrence Expression functions

Imposition time: `IT_SubClasTdEvAss`

〔The functions:

- `TIMEX_occurs`,
- `TIMEX_hasOccurred`,
- `TIMEX_timeSinceLastOccurrence`,
- `TIMEX_angleSinceLastOccurrence`,
- `TIMEX_modeActive`

shall only be used for an `TDEventOccurrenceExpressionFormula` applied to a `TDEventComplex`.]

[constr_4551] Use only Numericals in `TDEventOccurrenceExpression`

Imposition time: `IT_SubClasTdEvAss`

[The target data prototype of the instance references of `variable` and `argument` shall be `Numerical`.]

The `TDEventOccurrenceExpressionFormula` is specified such that it describes an *event* rather than a state. As a consequence the `TDEventOccurrenceExpressionFormula` shall ensure that a complex `TimingDescriptionEvent` *could* only occur at the occurrence time of one of the referenced `TimingDescriptionEvent`s.

[constr_4502] Use references only as function operands

Imposition time: `IT_SubClasTdEvAss`

[The references to model elements (e.g. the *timing event* reference targeting `TimingDescriptionEvent`) do have specific semantics. The usage of these references within the expression is *only* allowed as operand of the functions mentioned above.]

The example given below shows how to combine the functions introduced above in order to specify a `TDEventOccurrenceExpressionFormula` for a complex event called *ECX*. This complex event expresses that the two `dataElement`s *DE1* and *DE2* are becoming available at the required port *RequiredPort* of *Swc1* within an interval of maximum 0.5 ms *and* that the value of the `dataElement` *DE3* is larger than three (3) at this point in time.

Figure 2.35 sketches the AUTOSAR software component model of this example.

An SWC *Swc1* has a `RPortPrototype` called *RequiredPort*, and a `PPortPrototype` called *ProvidedPort*. Both ports are typed by a `SenderReceiverInterface` called *SenderReceiverInterface1*, and consist of three `dataElement`s: *DE1*, *DE2* and *DE3*¹.

¹one could also instead define three discrete `RPortPrototype`s typed by a `SenderReceiverInterface`, each with a single `dataElement`

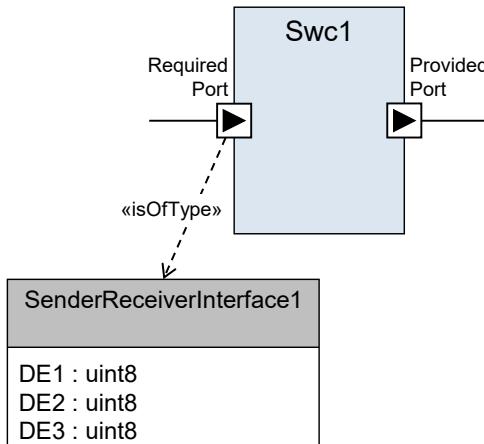


Figure 2.35: The SWC used by the Occurrence Expression Example

The corresponding timing model for the given example is shown in listing C.14. Since the timing is described for a software component in the Virtual Functional Bus view, the [VfbTiming](#) is used for specifying the corresponding timing model, namely the Virtual Functional Bus Timing View. This timing model shall only contain timing description events related to the Virtual Functional Bus as described in section [2.3.2.1](#).

The complex event *ECX* occurs when the following conditions are fulfilled:

Condition1 Either atomic timing event *E1* or *E2* shall occur. In this example, *E1* and *E2* are atomic timing events [TDEventVariableDataPrototype](#) which occur when the [VariableDataPrototypes](#) called *DE1* and *DE2* are received on [PortPrototype](#) called *Required Port* of the component called *Swc1*.

Condition2 The value of the [VariableDataPrototype](#) called *DE3* shall be greater than 3.

Condition3 The [VariableDataPrototypes](#) called *DE1* and *DE2* shall become available at the [required PortPrototype](#) called *RequiredPort* within a time interval of maximum 0.5 milliseconds.

The complex event *ECX* would be described by the following occurrence expression in [C.5.1](#)

Due to the first condition the complex event *ECX* can only occur when one of the atomic timing events *E1* or *E2* occurs at the point in time of evaluation and thus satisfies the principle that the occurrence expresses the semantics of an event and not a state.

Figure [2.38](#) shows a measurement of the event occurrences.

The corresponding AUTOSAR ARXML file fragment for the complex event *ECX* has the following appearance:

| | | | | |
|----------------------|---|--------------|-------------|--|
| Class | TDEventOccurrenceExpression | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescriptionEvents::TDEventOccurrenceExpression | | | |
| Note | This is used to specify a filter on the occurrences of TimingDescriptionEvents by means of a TDEventOccurrenceExpressionFormula. Filter criteria can be variable and argument values, i.e. the timing event only occurs for specific values, as well as the temporal characteristics of the occurrences of arbitrary timing events. | | | |
| Base | ARObject | | | |
| Aggregated by | TimingDescriptionEvent.occurrenceExpression | | | |
| Attribute | Type | Mult. | Kind | Note |
| argument | AutosarOperationArgumentInstance | * | aggr | An occurrence expression can reference an arbitrary number of OperationArgumentPrototypes in its expression. This association aggregates instance references to OperationArgumentPrototypes which can be referenced in the expression. |
| formula | TDEventOccurrenceExpressionFormula | 0..1 | aggr | This is the expression formula which is used to describe the occurrence expression. |
| mode | TimingModelInstance | * | aggr | An occurrence expression can reference an arbitrary number of TimingModelInstances in its expression. This association aggregates instance references to Mode Declaration which can be referenced in the expression. |
| variable | AutosarVariableInstance | * | aggr | An occurrence expression can reference an arbitrary number of VariableDataPrototypes in its expression. This association aggregates instance references to Variable DataPrototypes which can be referenced in the expression. |

Table 2.49: TDEventOccurrenceExpression

[constr_6879] Existence of [TDEventOccurrenceExpression.formula](#)
Imposition time: [IT_SubClasTdEvAss](#)

 [For each [TDEventOccurrenceExpression](#), the attribute [formula](#) shall exist]

| | | | | |
|----------------------|--|--------------|-------------|---|
| Class | «atpMixedString» TDEventOccurrenceExpressionFormula | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescriptionEvents::TDEventOccurrenceExpression | | | |
| Note | This is an extension of the FormulaExpression for the AUTOSAR Timing Extensions. A TDEventOccurrenceExpressionFormula provides the means to express the temporal characteristics of timing event occurrences in correlation with specific variable and argument values. The formal definition of the extended functions (ExtUnaryFunctions) is described in detail in the AUTOSAR Timing Extensions. | | | |
| Base | ARObject, FormulaExpression | | | |
| Aggregated by | TDEventOccurrenceExpression.formula | | | |
| Attribute | Type | Mult. | Kind | Note |
| argument | AutosarOperationArgumentInstance | 0..1 | ref | This is one particular argument value used in the expression formula. |
| event | TimingDescriptionEvent | 0..1 | ref | This is one particular timing description event used in the expression formula. |
| mode | TimingModelInstance | 0..1 | ref | This is one particular mode used in the expression formula. |
| variable | AutosarVariableInstance | 0..1 | ref | This is one particular variable value used in the expression formula. |

Table 2.50: TDEventOccurrenceExpressionFormula

| | | | | |
|----------------------|---|--------------|-------------|--|
| Class | AutosarVariableInstance | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescription Events::TDEventOccurrenceExpression | | | |
| Note | <p>This class represents a reference to a variable instance within AUTOSAR. This way it is possible to reference a variable instance in the occurrence expression formula. The variable instance can target to one of the following variables:</p> <ul style="list-style-type: none"> • a variable provided via a PortPrototype as whole • an element inside of a composite variable provided via a PortPrototype | | | |
| Base | ARObject , Identifiable , MultilanguageReferrable , Referrable | | | |
| Aggregated by | TDEventOccurrenceExpression.variable , TimingExtensionResource.timingVariable | | | |
| Attribute | Type | Mult. | Kind | Note |
| variableInstance | DataPrototype | 0..1 | iref | <p>This is the reference to the instanceRef definition.</p> <p>InstanceRef implemented by: VariableInComponent InstanceRef</p> |

Table 2.51: AutosarVariableInstance

[constr_6880] Existence of [AutosarVariableInstance.variableInstance](#)

Imposition time: [IT_SubClasTdEvAss](#)

[For each [AutosarVariableInstance](#), the reference in the role [variableInstance](#) shall exist]

| | | | | |
|-----------------------------|--|--------------|-------------|---|
| Class | AutosarOperationArgumentInstance | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescription Events::TDEventOccurrenceExpression | | | |
| Note | <p>This class represents a reference to an argument instance. This way it is possible to reference an argument instance in the occurrence expression formula. The argument instance can target to one of the following arguments:</p> <ul style="list-style-type: none"> • a whole argument used in an operation of a PortPrototype with ClientServerInterface • an element inside of a composite argument used in an operation of a PortPrototype with ClientServer Interface | | | |
| Base | ARObject , Identifiable , MultilanguageReferrable , Referrable | | | |
| Aggregated by | TDEventOccurrenceExpression.argument , TimingExtensionResource.timingArgument | | | |
| Attribute | Type | Mult. | Kind | Note |
| operation Argument Instance | DataPrototype | 0..1 | iref | <p>This is the reference to the instanceRef definition.</p> <p>InstanceRef implemented by: OperationArgumentInComponent InstanceRef</p> |

Table 2.52: AutosarOperationArgumentInstance

[constr_6881] Existence of [AutosarOperationArgumentInstance.operationArgumentInstance](#)

Imposition time: [IT_SubClasTdEvAss](#)

[For each [AutosarOperationArgumentInstance](#), the reference in the role [operationArgumentInstance](#) shall exist]

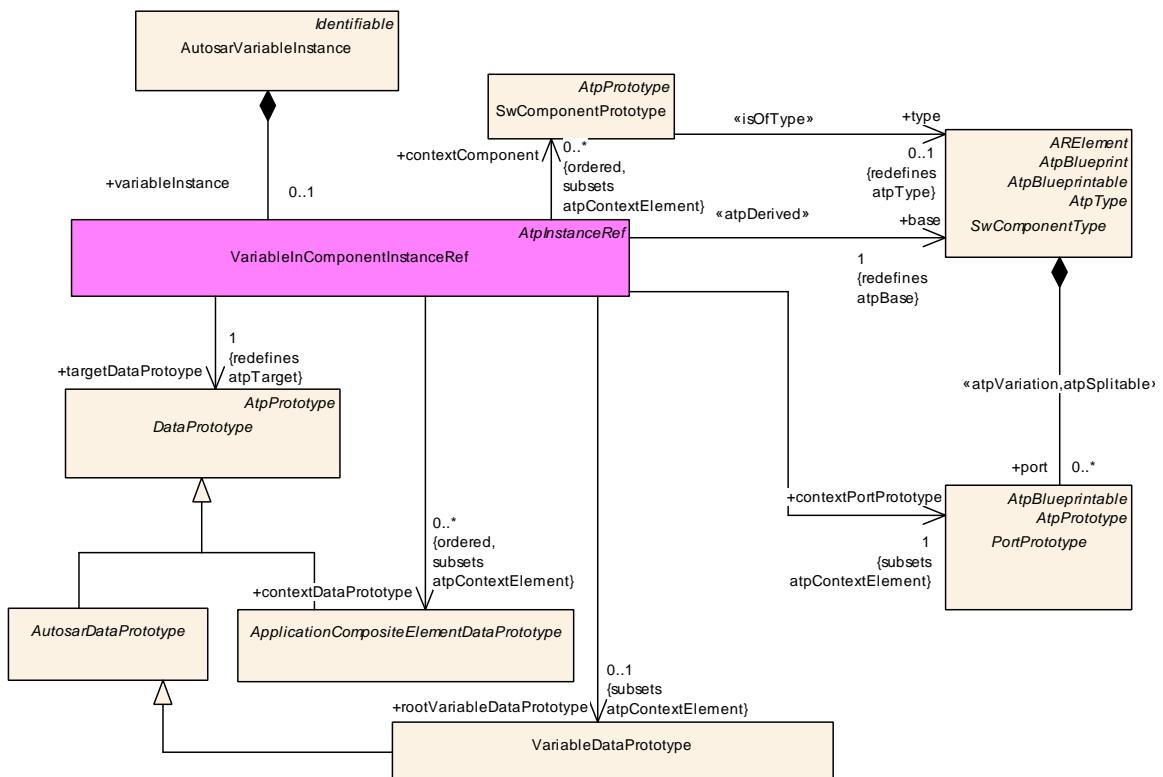


Figure 2.36: The required context information to reference a variable instance within AUTOSAR.

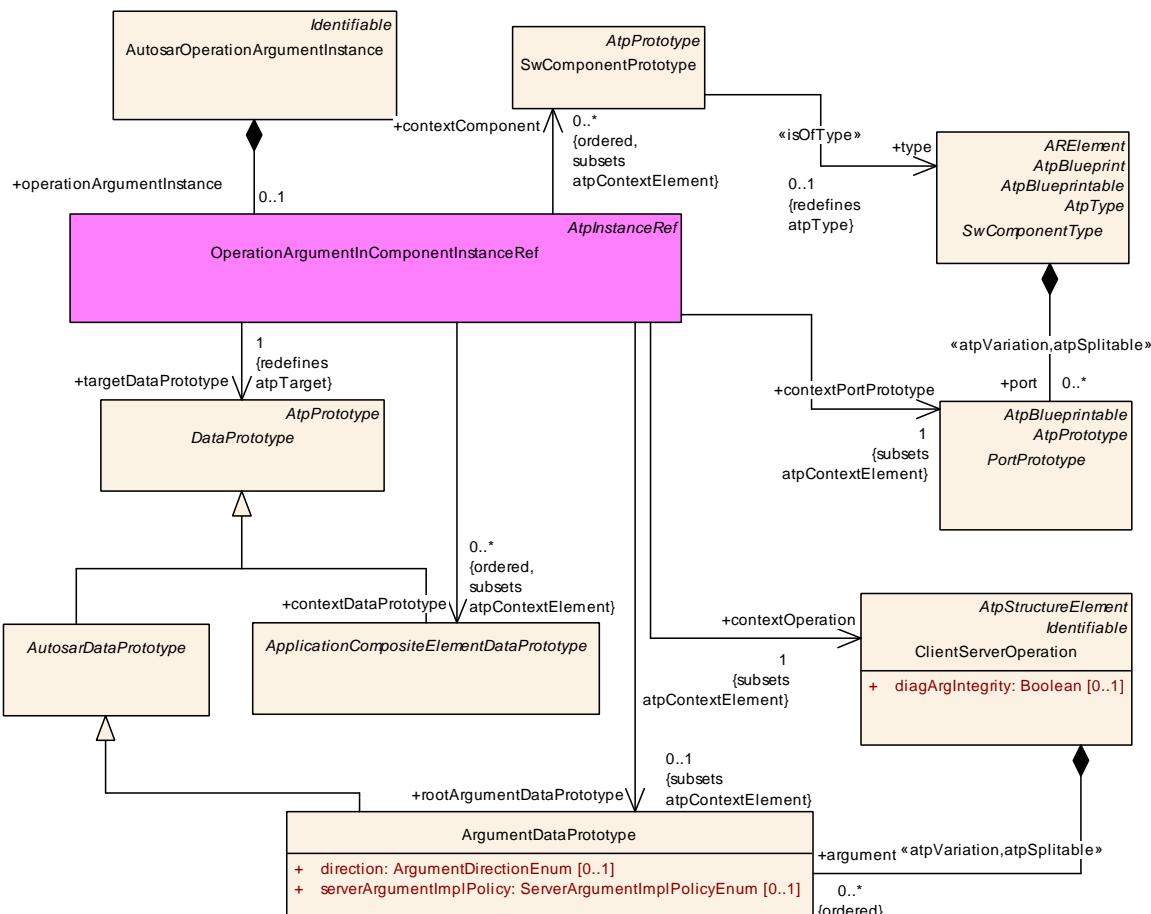


Figure 2.37: The required context information to reference an operation argument instance within AUTOSAR.

2.3.2.8 Occurrence Expression Language Syntax

The occurrence expression language is based on the syntax of the formula language defined in the Generic Structure Template [8]. It extends the language by additional functions and additional references to model elements. In the following, the implications of the extensions to the syntax are presented based on the grammar definition.

Note: The grammar defined for the formula language is not part of the listing below. It presents only the timing specific extensions of the formula language and the enhanced functions and references.

2.3.2.8.1 Interpreting an Occurrence Expression

Based on the specification mechanism described in the previous sections it is possible to use the occurrence expression formula to refine the timing specification to the intended precision. This section describes how such an occurrence expression has to

be interpreted. The duty of the interpreter is to determine the occurrences of the [TimingDescriptionEvent](#) for which the occurrence expression is defined. This is done in two ways, depending on whether the occurrence expression is used as a content filter or as a complex event.

2.3.2.8.1.1 Interpreting a Content Filter

In this case, the occurrence expression is defined for an atomic event. Only the unary timing function *TIMEX_value(<reference to argument or variable>)* is allowed to be used for the content filter. On each occurrence of the atomic event the interpreter checks whether the content filter defined by the expression is fulfilled. This is done by evaluating the function *TIMEX_value* based on its operand type:

AutosarVariableInstance the value of the referenced variable is evaluated at the point in time the atomic event occurs.

AutosarOperationArgumentInstance the value of the referenced argument is evaluated at the point in time the atomic event occurs.

[constr_4552] Restricted usage of [AutosarVariableInstance](#) for Content Filter

Imposition time: [IT_SubClasTdEvAss](#)

[If a content filter is defined for an atomic event then references to [AutosarVariableInstances](#) are only allowed if the atomic event is of type [TDEventVariableDataPrototype](#). Only if such an atomic event occurs, the value of the variables can be evaluated. Thus, also the scope of the atomic event shall be the same as the [AutosarVariableInstance](#), meaning that they shall point to the same [VariableDataPrototype](#).]

[constr_4503] Restricted usage of [AutosarOperationArgumentInstance](#) for Content Filter

Imposition time: [IT_SubClasTdEvAss](#)

[If a content filter is defined for an atomic event then references to [AutosarOperationArgumentInstances](#) are only allowed if the atomic event is of type [TDEventToOperation](#). Only if such an atomic event occurs, the value of the operation arguments can be evaluated. Thus, also the scope of the atomic event shall be the same as the [AutosarOperationArgumentInstance](#), meaning that they shall point to the same [ClientServerOperation](#). Finally, references to an [AutosarOperationArgumentInstance](#) with argument direction "out" are only allowed, if the atomic event of type [TDEventOperation](#) refers either to the point in time when the operation call response has been sent (TD-EVENT-OPERATION-TYPE=OPERATION-CALL-RESPONSE-SENT) or to the point in time when the operation call response has been received (TD-EVENT-OPERATION-TYPE=OPERATION-CALL-RESPONSE-RECEIVED).]

2.3.2.8.1.2 Interpreting a Complex Event

In this case, the occurrence expression is defined for a complex event. All features of the occurrence expression language can be used for this expression type. At a specific point in time t , the interpreter evaluates the expression to determine if the complex event has occurred.

Considering the occurrence expression defined for the example given in Section 2.3.2.7.1, the interpreter "implements" a function $ECX(t)$ which returns TRUE, if the complex event ECX occurs at time t :

```
ECX(t) =
( TIMEX_occurs( t, /Example/Expression/E1 )
  || TIMEX_occurs( t, /Example/Expression/E2 ) )
&& TIMEX_value( t, /Example/Expression/ECX/avi_DE3 ) > 3
&& abs( TIMEX_timeSinceLastOccurrence( t, /Example/Expression/E1 ) -
  TIMEX_timeSinceLastOccurrence( t, /Example/Expression/E2 ) ) <= 0.0005
```

The expression shall only be evaluated at occurrence times of $E1$ or $E2$, because only then the complex event ECX can occur and the expression can return TRUE.

As shown in the sketched trace in Figure 2.38 the timing description events called $E1$ and $E2$ occur at different times. On the left hand side of this figure the two events occur within a time interval of 0.0005 seconds. The point in time the given occurrence expression is evaluated is the point in time the event $E2$ occurs. The result of the occurrence expression at this point in time, $t_{evaluate}$ respectively t_{E2} , is TRUE. On the right hand side of this figure the two events do not occur within a time interval of 0.0005 seconds. The point in time the given occurrence expression is evaluated is the point in time the event $E1$ occurs. The result of the occurrence expression at this point in time, $t_{evaluate}$ respectively t_{E1} , is FALSE.

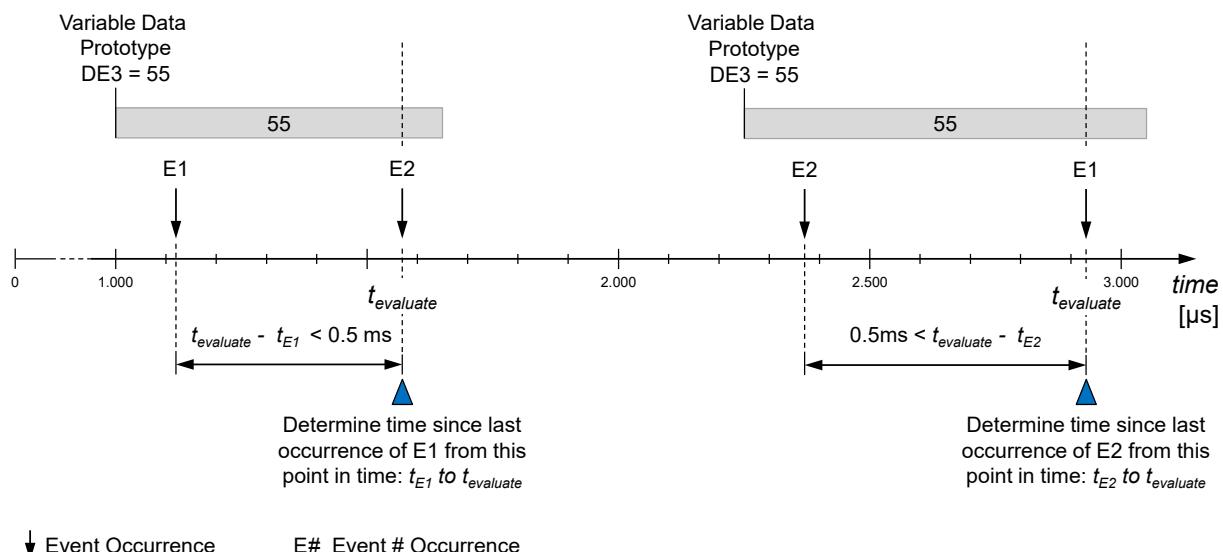


Figure 2.38: Trace showing various occurrences of the timing description events $E1$ and $E2$, as well as the value of the variable $DE3$.

Based on the several functions provided by the occurrence expression language, the interpreter requires the following information from the system:

- the value of a referenced `AutosarOperationArgumentInstance` at time t .
- the value of a referenced `AutosarVariableInstance` at time t .
- the occurrences of a referenced `TimingDescriptionEvent` at time t and before.

There are different ways to gather the required information:

- Model analysis and simulation: In a deterministic system environment, occurrences of `TimingDescriptionEvent`s can be determined offline, for example the point in time a frame will be transmitted in the static segment of a FlexRay network.
- Target trace: The required information can be gathered from a running system by recording the points in time a `TimingDescriptionEvent` has occurred. For example, an ECU trace for `TDEventSwcInternalBehavior` may contain a marker for each point in time a `RunnableEntity` is activated, started or terminated.

If the interpreter has the required information as input, the different functions provided by the occurrence expression language can be interpreted as follows:

- `TIMEX_value(t, <reference to an AutosarVariableInstance>)` returns the variable value at time t .
- `TIMEX_value(t, <reference to an AutosarOperationArgumentInstance>)` returns the operation argument value at time t .
- `TIMEX_occurs(t, <reference to a TimingDescriptionEvent>)` returns TRUE (or 1) if the referenced event has occurred at time t , else it returns FALSE (or 0).
- `TIMEX_hasOccurred(t, <reference to a TimingDescriptionEvent>)` returns TRUE (or 1) if the referenced event has occurred *at least once* before or at time t .
- `TIMEX_timeSinceLastOccurrence(t, <reference to a TimingDescriptionEvent>)` returns the time difference between t and the point in time of the last occurrence of the referenced event. The unit of time is seconds.
- `TIMEX_angleSinceLastOccurrence(t, <reference to a TimingDescriptionEvent>)` returns the angle difference between t and the point in time of the last occurrence of the referenced event. The unit of angle is degree.
- `TIMEX_modeActive(t, <reference to a TimingModeInstance>)` returns TRUE (or 1) if the referenced mode is active at time t , else it returns FALSE (or 0).

2.3.2.9 Time Base Referencing for Timing Description Events

In certain use-cases, a [TimingDescriptionEvent](#) may reference an optional [TimingDescriptionEvent.clockReference](#) to provide a relation to an *actual* time base, this concept is explained in section [3.2](#).

2.4 TimingConstraint

TIMEX [TimingConstraints](#) may be used two fold:

1. as a restriction for the timing behavior of the system (e.g. minimum (maximum) latency bound for a certain event sequence). In this case, a timing constraint is a functional requirement which the system shall fulfill as a basis for testing.
2. as a guarantee for the timing behavior of the system. In this case, the system developer guarantees that the system has a certain behavior with respect to timing (e.g. a timing event is guaranteed to occur periodically with a certain maximum variation). For example, in the context of a performance specification.

The application of [TimingConstraints](#) are contextual and are applied to constrain [TimingDescriptionEvents](#), [TimingDescriptionChains](#) or [ExecutableEntity](#)s.

2.4.1 TimingConstraints on TimingDescriptionEvents

A [TimingDescriptionEvent](#) is used to describe that in a system, specific events occur and also at which locations in this system the occurrences are observed. Generally a [TimingConstraint](#), when applied to [TimingDescriptionEvent](#), imposes a restriction/expectation on the occurrences of an event in a given scope/context with a temporal restriction.

An [EventTriggeringConstraint](#) specifies the way an event occurs in the temporal space. TIMEX provides means to specify periodic ([PeriodicEventTriggering](#)) and sporadic ([SporadicEventTriggering](#)) occurrences, as well as event occurrences that follow a specific pattern ([BurstPatternEventTriggering](#), [ConcretePatternEventTriggering](#), and [ArbitraryEventTriggering](#) pattern).

In [2.39](#) the differing sub-classes of [TimingConstraints](#) on [TimingDescriptionEvents](#) are shown.

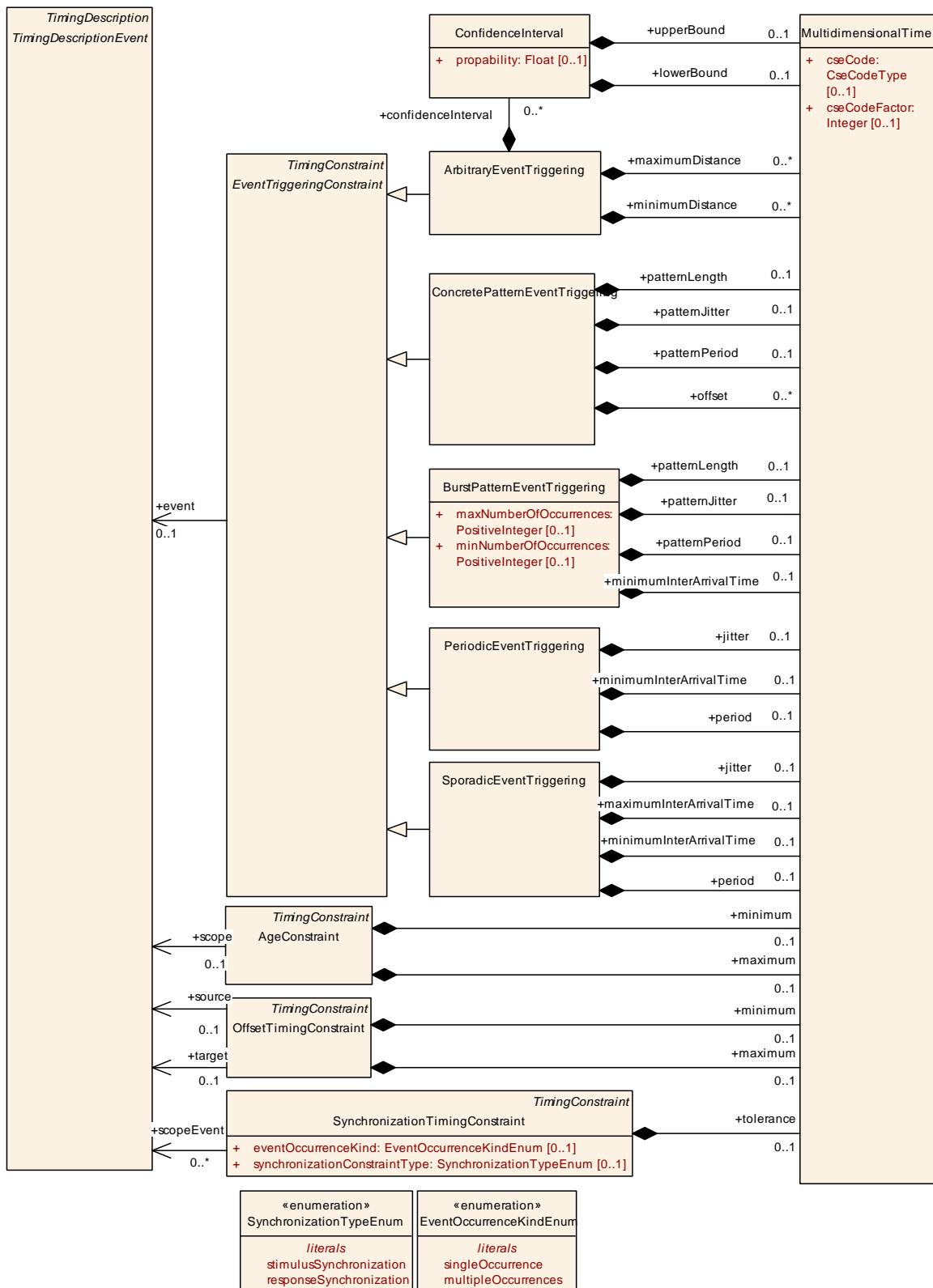


Figure 2.39: **TimingConstraints on TimingDescriptionEvents**

2.4.1.1 SynchronizationTimingConstraint

In some cases the complete event chains are not entirely known, or not available in the scope of the model, at the point in time the timing constraint shall be specified. For this purpose, TIMEX allows the specification of synchronization constraints on events. In this case, the events referenced by the constraint are related implicitly, because they have a common stimulus (in case of constraint type `responseSynchronization` or a common response (in case of constraint type `stimulusSynchronization` not known yet, or not available in the scope of the model.

At a later stage during the development, when the refined software architecture exposes the complete event chains (e.g. because the common stimulus gets known), the respective event chains shall be specified and associated with a `SynchronizationTimingConstraint` on event chains (see 2.4.2.1) in order to refine the previously defined `SynchronizationTimingConstraint` on events.

[TPS_TIMEX_00006] `SynchronizationTimingConstraint` specifies synchronicity constraints

Upstream requirements: RS_TIMEX_00001, RS_TIMEX_00002, RS_TIMEX_00007, RS_TIMEX_00008, RS_TIMEX_00017

〔The element `SynchronizationTimingConstraint` is used to specify a synchronization constraint among the occurrences of two or more `TimingDescriptionEvents`.〕

The parameters are shown in use in Figure 2.40.

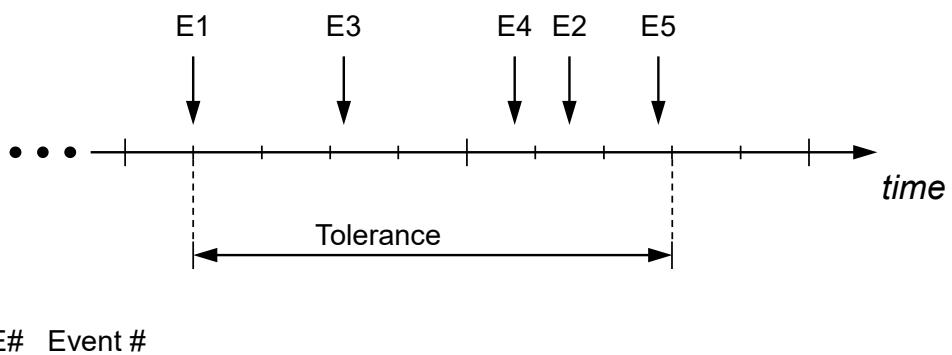


Figure 2.40: Parameter characterizing the Synchronization Constraint

[constr_4513] `SynchronizationTimingConstraint` shall reference at least two events

Imposition time: IT_SubClasTdEvAss

〔In the case, that the `SynchronizationTimingConstraint` is imposed on events then at least two (2) timing description events shall be referenced.〕

[constr_4520] Specifying attribute `synchronizationConstraintType`

Imposition time: `IT_SubClasTdEvAss`

〔The attribute `synchronizationConstraintType` shall be specified if the `SynchronizationTimingConstraint` is imposed on events.〕

[constr_6846] Existence of `SynchronizationTimingConstraint.synchronizationConstraintType`

Imposition time: `IT_SubClasTdEv`

〔For each `SynchronizationTimingConstraint`, the attribute `synchronizationConstraintType` shall exist〕

[constr_6847] Existence of `SynchronizationTimingConstraint.tolerance`

Imposition time: `IT_SubClasTdEv`

〔For each `SynchronizationTimingConstraint`, the attribute `tolerance` shall exist〕

2.4.1.2 EventTriggeringConstraint

The `EventTriggeringConstraint` is an abstract base class representing a number sub-classes of concrete types (patterns) of event reporting. Regardless of the concrete meta-class, the `EventTriggeringConstraint` needs an `event` as its scope.

[TPS_TIMEX_00003] `EventTriggeringConstraint` specifies occurrence behavior respectively model

Upstream requirements: `RS_TIMEX_00001`, `RS_TIMEX_00002`, `RS_TIMEX_00006`, `RS_TIMEX_00008`

〔The element `EventTriggeringConstraint` is an abstract meta-class to specify the particular occurrences of a given `TimingDescriptionEvent`.〕

| | |
|----------------------|---|
| Class | <code>EventTriggeringConstraint</code> (abstract) |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint::EventTriggeringConstraint |
| Note | <p>Describes the occurrence behavior of the referenced timing event.</p> <p>The occurrence behavior can only be determined when a mapping from the timing events to the implementation can be obtained. However, such an occurrence behavior can also be described by the modeler as an assumption or as a requirement about the occurrence of the event.</p> |
| Base | <code>ARObject</code> , <code>Identifiable</code> , <code>MultilanguageReferrable</code> , <code>Referrable</code> , <code>TimingConstraint</code> , <code>Traceable</code> |
| Subclasses | <code>ArbitraryEventTriggering</code> , <code>BurstPatternEventTriggering</code> , <code>ConcretePatternEventTriggering</code> , <code>PeriodicEventTriggering</code> , <code>SporadicEventTriggering</code> |
| Aggregated by | <code>TimingExtension.timingGuarantee</code> , <code>TimingExtension.timingRequirement</code> |





| Class | EventTriggeringConstraint (abstract) | | | |
|-----------|--------------------------------------|-------|------|-----------------------------|
| Attribute | Type | Mult. | Kind | Note |
| event | TimingDescriptionEvent | 0..1 | ref | The referenced timing event |

Table 2.53: EventTriggeringConstraint

[constr_6818] Existence of `EventTriggeringConstraint.event`

Imposition time: IT_SubClasTdEvAss

〔For each `EventTriggeringConstraint`, the reference in the role `event` shall exist〕

2.4.1.2.1 PeriodicEventTriggering

[TPS_TIMEX_00010] `PeriodicEventTriggering` specifies periodic occurrences of events

Upstream requirements: RS_TIMEX_00001, RS_TIMEX_00002, RS_TIMEX_00006, RS_TIMEX_00008, RS_TIMEX_00015

〔The element `PeriodicEventTriggering` is used to specify the characteristics of a `TimingDescriptionEvent` which occurs periodically.〕

| Class | PeriodicEventTriggering | | | |
|-------------------------|--|-------|------|--|
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint::EventTriggeringConstraint | | | |
| Note | Describes the behavior of an event with a strict periodic occurrence pattern, given by <code>period</code> . Additionally, it is possible to soften the strictness of the periodic occurrence behavior by specifying a <code>jitter</code> , so that there can be a deviation from the <code>period</code> up to the size of the <code>jitter</code> . | | | |
| Base | ARObject, <code>EventTriggeringConstraint</code> , <code>Identifiable</code> , <code>MultilanguageReferrable</code> , <code>Referrable</code> , <code>Timing Constraint</code> , <code>Traceable</code> | | | |
| Aggregated by | <code>TimingExtension.timingGuarantee</code> , <code>TimingExtension.timingRequirement</code> | | | |
| Attribute | Type | Mult. | Kind | Note |
| jitter | MultidimensionalTime | 0..1 | aggr | The maximum deviation of the periodic event occurrence. Tags: xml.sequenceOffset=20 |
| minimumInterArrivalTime | MultidimensionalTime | 0..1 | aggr | The minimum time distance between subsequent consecutive occurrences of the associated event. If the <code>minimumInterArrivalTime</code> is less than the <code>period</code> minus the <code>jitter</code> , then the <code>minimumInterArrivalTime</code> has no effect on the properties of the constraint. Tags: xml.sequenceOffset=10 |
| period | MultidimensionalTime | 0..1 | aggr | The periodic distance between subsequent occurrences of the event. Tags: xml.sequenceOffset=30 |

Table 2.54: PeriodicEventTriggering

Let t_n be the point-in-time of the n -th occurrence of the event. A [PeriodicEventTriggering](#) Constraint is satisfied if, and only if, at least one reference point-in-time $t_{reference}$ exists such that for every occurrence of the event at t_n the following holds true:

$$t_{reference} + (n - 1)period \leq t_n \leq t_{reference} + (n - 1)period + jitter$$

and for all of those event occurrences the minimum distance shall be less than or equal to [minimumInterArrivalTime](#).

$$\exists t_{reference} \mid \forall n : t_{reference} + (n - 1)period \leq t_n \leq t_{reference} + (n - 1)period + jitter$$

AND $\forall n : t_{n+1} - t_n \leq minimumInterArrivalTime$

Figure 2.41 illustrates the parameters of the [PeriodicEventTriggering](#). The upper part of this figure shows the case that the value of [jitter](#) is less than the value of the parameter [period](#); whereas the lower part of this figure shows the $jitter \geq period$.

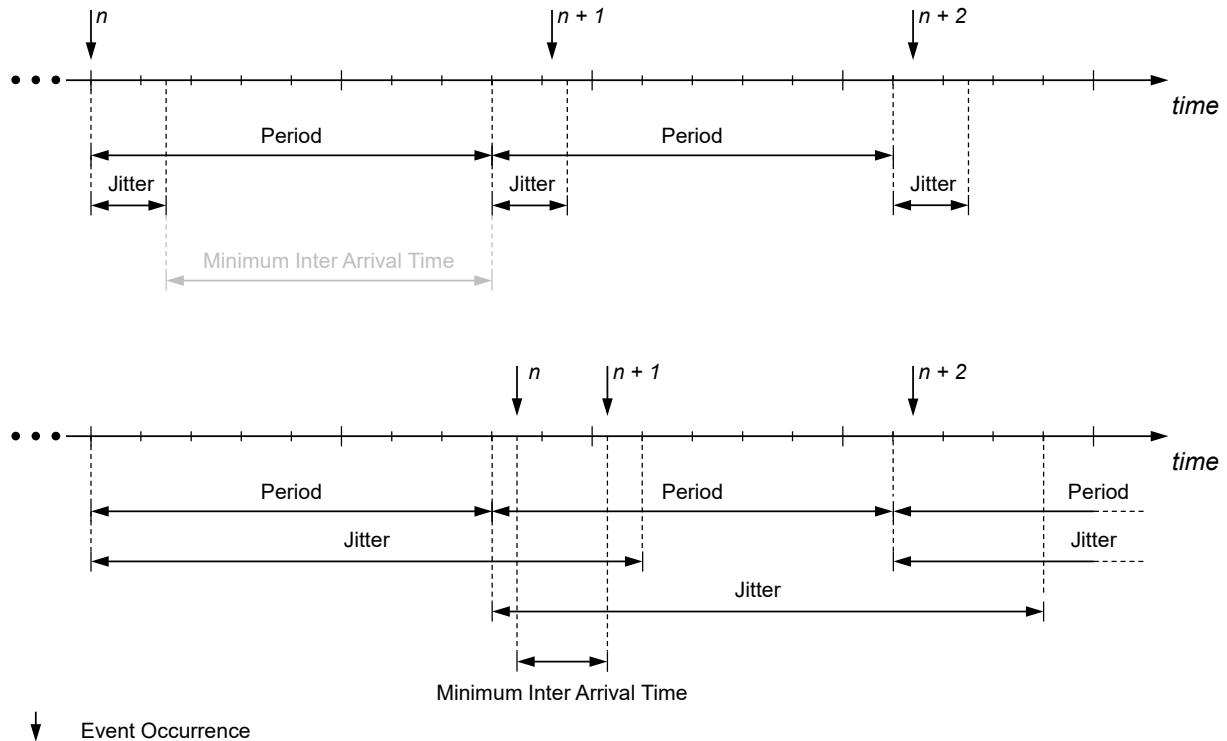


Figure 2.41: Parameters characterizing the PeriodicEventTriggering

Example: A [PeriodicEventTriggering](#) Constraint is specified with the following parameters: $period=6\text{ms}$ and $jitter=2\text{ms}$. In other words, one imposes a timing constraint on an event to occur every 6ms and specifies that a deviation of 2ms is tolerable.

In addition, it is assumed that the [minimumInterArrivalTime=1ms](#) and therefore has no impact on the timing of the event's occurrences. This timing constraint is shown in Figure 2.42. The repeating gray-colored rectangles in this figure indicate the time intervals during which the event may occur; i.e. they demarcate the subsequent time intervals the event is expected to occur.

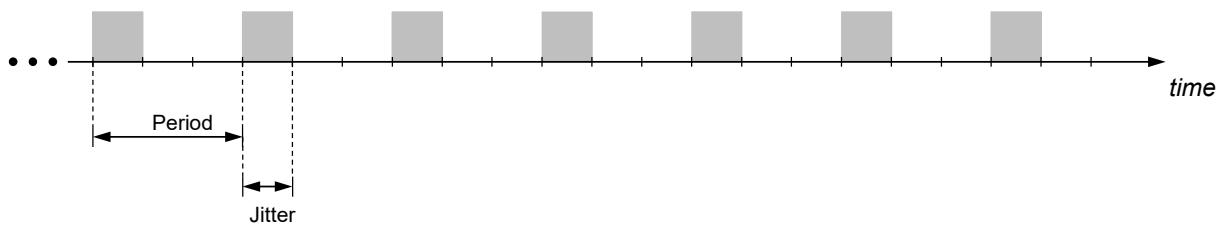


Figure 2.42: Example of a [PeriodicEventTriggering](#) Constraint

The following figures show various event occurrences recorded during the observation of a system subject to analysis. The time interval for the observation is given by $t_{end-observation} - t_{start-observation}$. In the given [example](#), the system is observed for a period of 33.6ms.

The subsequent event occurrences shown in Figure 2.43 satisfy the given [PeriodicEventTriggering](#) constraint, because all occurrences of the event observed during the observation time interval happen in their corresponding time interval given by [period](#) and [jitter](#).

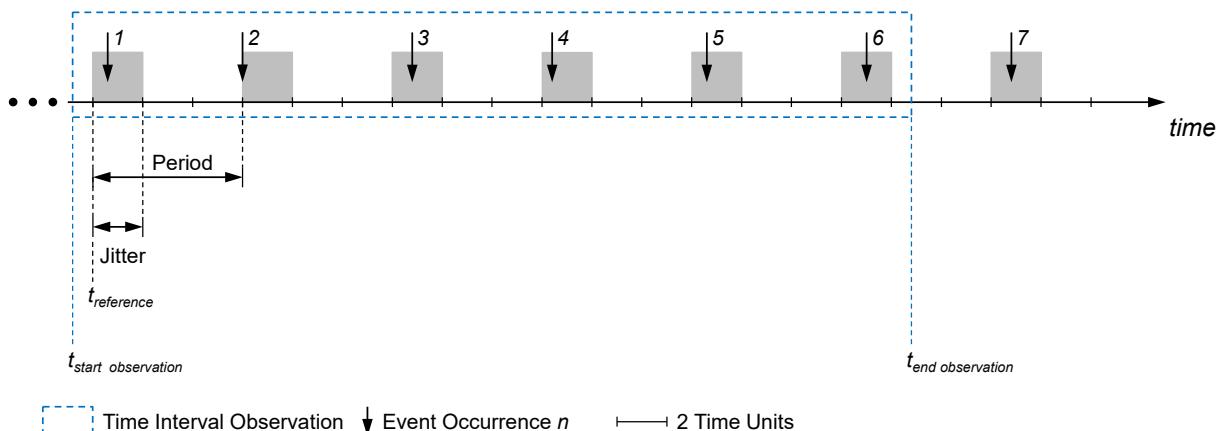


Figure 2.43: Event occurrences satisfying the given [PeriodicEventTriggering](#) Constraint shown in the [example](#)

The subsequent event occurrences shown in Figure 2.44 satisfy the given periodic event triggering constraint, because all occurrences of the event observed during the observation time interval happen in their corresponding time interval given by [period](#) and [jitter](#). In contrast to the example shown in Figure 2.43 the reference point-in-time is another one.

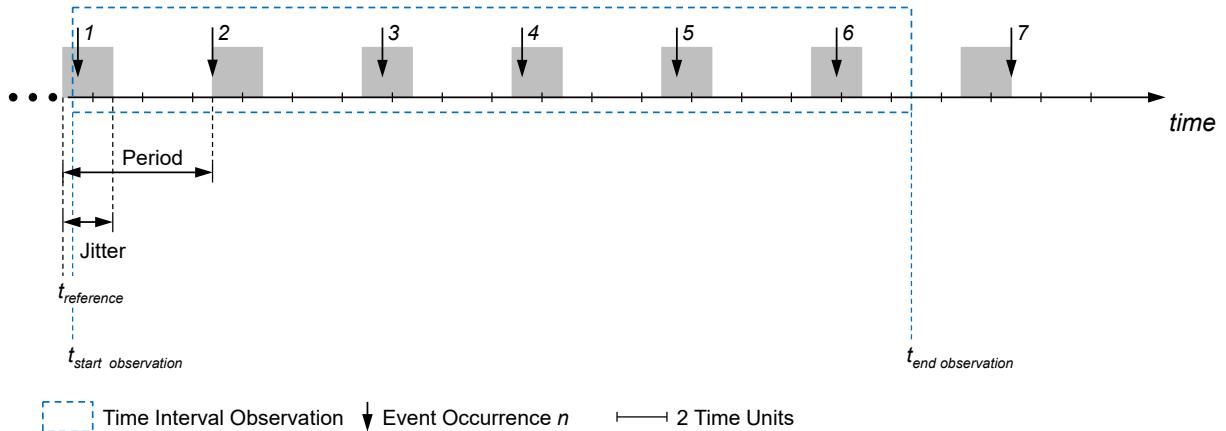


Figure 2.44: Event occurrences satisfying the given [PeriodicEventTriggering](#) Constraint shown in the [example](#), but with another reference point-in-time $t_{reference}$.

The subsequent event occurrences shown in Figure 2.45 violate the given [PeriodicEventTriggering](#) constraint, because the fifth occurrence of the event does not happen in its corresponding time interval given by [period](#) and [jitter](#). In other words, there does not exist a reference point-in-time that ensures that all occurrences of the event observed during the observation time interval happen in their corresponding time interval given by [period](#) and [jitter](#). This results in a violation of the parameters [period](#) and [jitter](#).

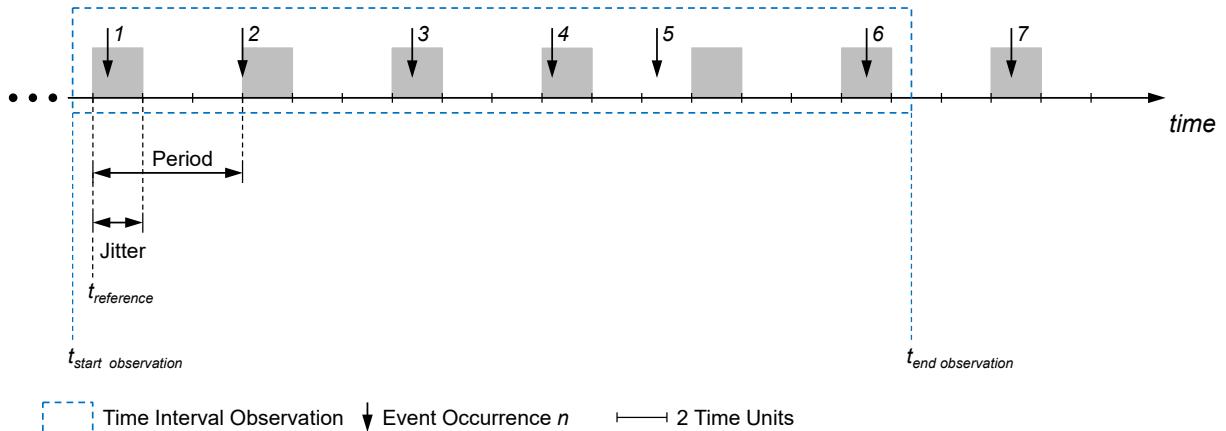


Figure 2.45: Event occurrences violating the given [PeriodicEventTriggering](#) Constraint shown in the [example](#) at the beginning of this subsection.

The subsequent event occurrences shown in Figure 2.46 violate the given periodic event triggering constraint, because the fourth occurrence of the event does not happen in its corresponding time interval given by [period](#) and [jitter](#). In other words, the fourth occurrence of the event happens in the time interval the fifth occurrence of the event happens and therefore violates the specified [jitter](#).

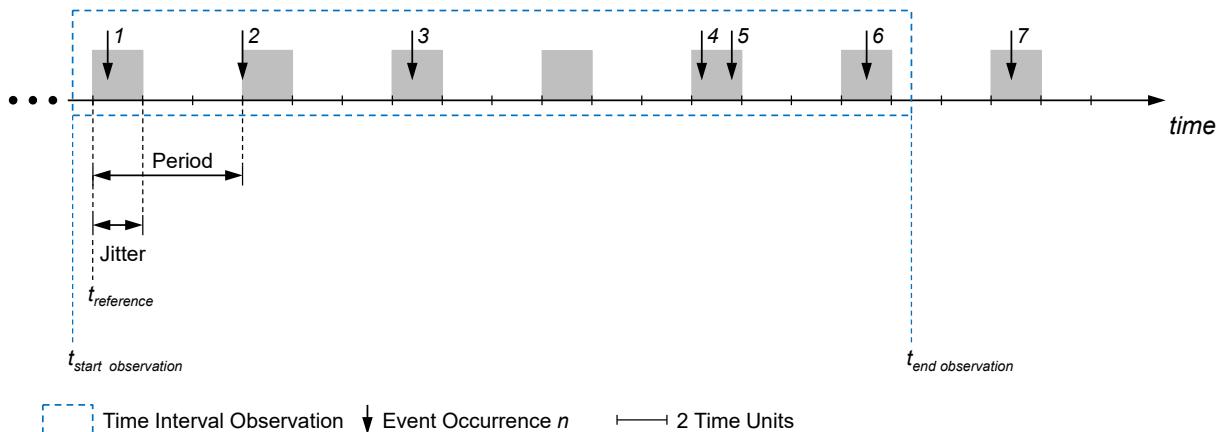


Figure 2.46: Event occurrences satisfying the given `PeriodicEventTriggering` Constraint shown in the example

[constr_4543] Maximum value of `minimumInterArrivalTime`

Imposition time: IT_SubClasTdEvAss

「The `minimumInterArrivalTime` shall be \leq the `period`.」

[constr_6819] Existence of `PeriodicEventTriggering.jitter`

Imposition time: IT_SubClasTdEvAss

「For each `PeriodicEventTriggering`, the attribute `jitter` shall exist」

[constr_6820] Existence of `PeriodicEventTriggering.minimumInterArrivalTime`

Imposition time: IT_SubClasTdEvAss

「For each `PeriodicEventTriggering`, the attribute `minimumInterArrivalTime` shall exist」

[constr_6821] Existence of `PeriodicEventTriggering.period`

Imposition time: IT_SubClasTdEvAss

「For each `PeriodicEventTriggering`, the attribute `period` shall exist」

2.4.1.2.2 SporadicEventTriggering

In contrast to a `PeriodicEventTriggering`, a `SporadicEventTriggering` permits events which may not necessarily occur. It adds one additional parameter: the `maximumInterArrivalTime` to specify the largest possible time distance between two `TimingDescriptionEvent` occurrences.

[TPS_TIMEX_00011] `SporadicEventTriggering` specifies sporadic occurrences of events

Upstream requirements: [RS_TIMEX_00001](#), [RS_TIMEX_00002](#), [RS_TIMEX_00006](#), [RS_TIMEX_00008](#)

[The element `SporadicEventTriggering` is used to specify the characteristics of a `TimingDescriptionEvent` which occurs sporadically.]

| Class | <code>SporadicEventTriggering</code> | | | |
|-------------------------|---|-------|------|---|
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint::EventTriggeringConstraint | | | |
| Note | Describes the behavior of an event which occurs occasionally or singularly. | | | |
| Base | ARObject, EventTriggeringConstraint , Identifiable , MultilanguageReferrable , Referrable , Timing Constraint , Traceable | | | |
| Aggregated by | TimingExtension.timingGuarantee , TimingExtension.timingRequirement | | | |
| Attribute | Type | Mult. | Kind | Note |
| jitter | MultidimensionalTime | 0..1 | aggr | The maximum deviation of the sporadic event occurrence. Jitter=max nthPeriod - standardPeriod Tags: xml.sequenceOffset=30 |
| maximumInterArrivalTime | MultidimensionalTime | 0..1 | aggr | The maximum time distance between two consecutive (subsequent) occurrences of the associated event. Tags: xml.sequenceOffset=20 |
| minimumInterArrivalTime | MultidimensionalTime | 0..1 | aggr | The minimum time distance between two consecutive (subsequent) occurrences of the associated event. Tags: xml.sequenceOffset=10 |
| period | MultidimensionalTime | 0..1 | aggr | The periodic distance between subsequent occurrences of the event. Tags: xml.sequenceOffset=40 |

Table 2.55: `SporadicEventTriggering`

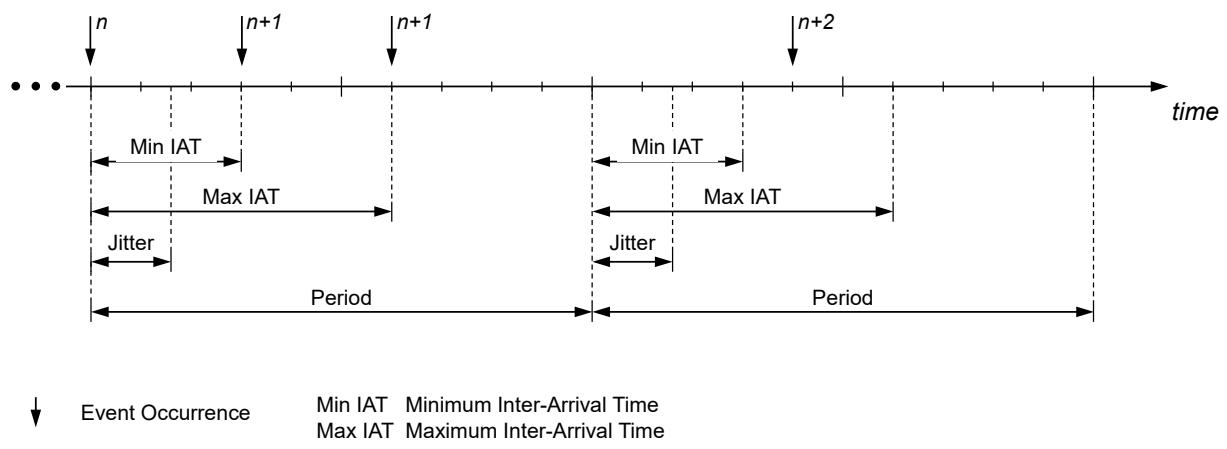


Figure 2.47: Parameters characterizing the `SporadicEventTriggering` Constraint

[constr_6822] Existence of `SporadicEventTriggering.maximumInterArrivalTime`

Imposition time: `IT_SubClasTdEvAss`

〔For each `SporadicEventTriggering`, the attribute `maximumInterArrivalTime` shall exist〕

[constr_6823] Existence of `SporadicEventTriggering.minimumInterArrivalTime`

Imposition time: `IT_SubClasTdEvAss`

〔For each `SporadicEventTriggering`, the attribute `minimumInterArrivalTime` shall exist〕

2.4.1.2.3 ConcretePatternEventTriggering

[TPS_TIMEX_00012] `ConcretePatternEventTriggering` specifies concrete pattern of occurrences of events

Upstream requirements: `RS_TIMEX_00001`, `RS_TIMEX_00002`, `RS_TIMEX_00006`, `RS_TIMEX_00008`

〔The element `ConcretePatternEventTriggering` is used to specify the characteristics of a `TimingDescriptionEvent` which occurs as a concrete pattern.〕

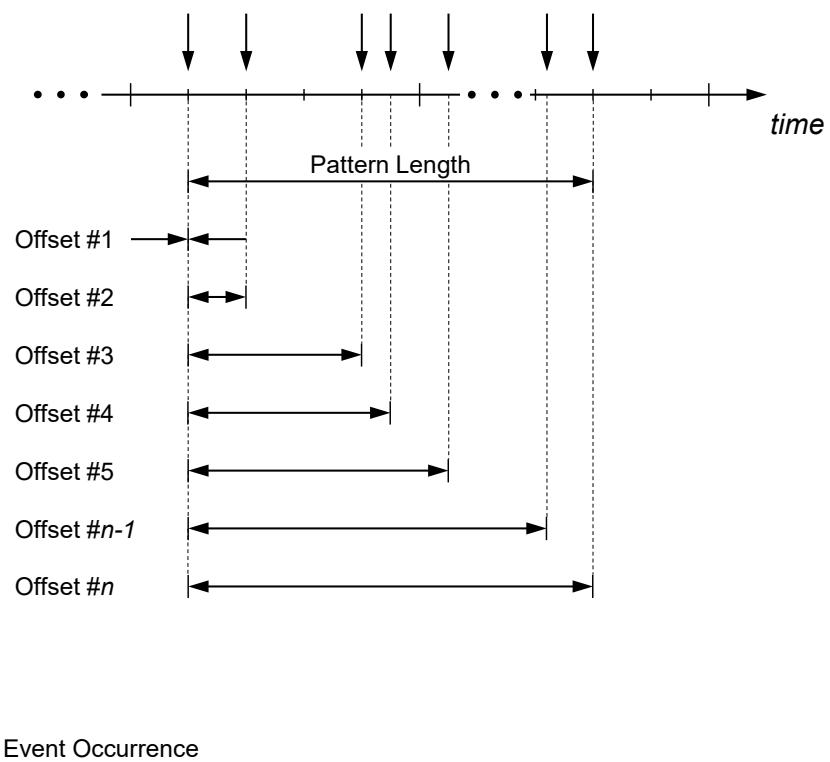
| Class | <code>ConcretePatternEventTriggering</code> | | | |
|---------------|---|-------|------|---|
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint::EventTriggeringConstraint | | | |
| Note | Describes the behavior of an event that occurs according to a precisely known pattern. | | | |
| Base | <code>ARObject</code> , <code>EventTriggeringConstraint</code> , <code>Identifiable</code> , <code>MultilanguageReferrable</code> , <code>Referrable</code> , <code>Timing Constraint</code> , <code>Traceable</code> | | | |
| Aggregated by | <code>TimingExtension.timingGuarantee</code> , <code>TimingExtension.timingRequirement</code> | | | |
| Attribute | Type | Mult. | Kind | Note |
| offset | MultidimensionalTime | * | agr | <p>The offset for each occurrence of the event in the specified time interval. A list of point-in-times in the time interval given by the parameter <code>patternLength</code> at which the event occurs.</p> <p>Tags: <code>xml.name=TIME-VALUE</code> <code>xml.roleElement=true</code> <code>xml.sequenceOffset=10</code> <code>xml.typeElement=false</code> </p> |
| patternJitter | MultidimensionalTime | 0..1 | agr | The maximum deviation of the time interval's starting point from the beginning of the given period. This parameter is only applicable in conjunction with the parameter <code>patternPeriod</code> . |



| Class | ConcretePatternEventTriggering | | | |
|---------------|--------------------------------|------|------|---|
| patternLength | MultidimensionalTime | 0..1 | aggr | The duration of the time interval within which the event repeatedly occurs. The event occurs at concrete points in time within the given time interval. Tags: xml.sequenceOffset=20 |
| patternPeriod | MultidimensionalTime | 0..1 | aggr | The time distance between the beginnings of subsequent repetitions of the given concrete pattern. |

Table 2.56: ConcretePatternEventTriggering

Figure 2.48 and Figure 2.49 illustrate the parameters of the [ConcretePatternEventTriggering](#) in use.


Figure 2.48: Parameters characterizing the [ConcretePatternEventTriggering](#) Constraint

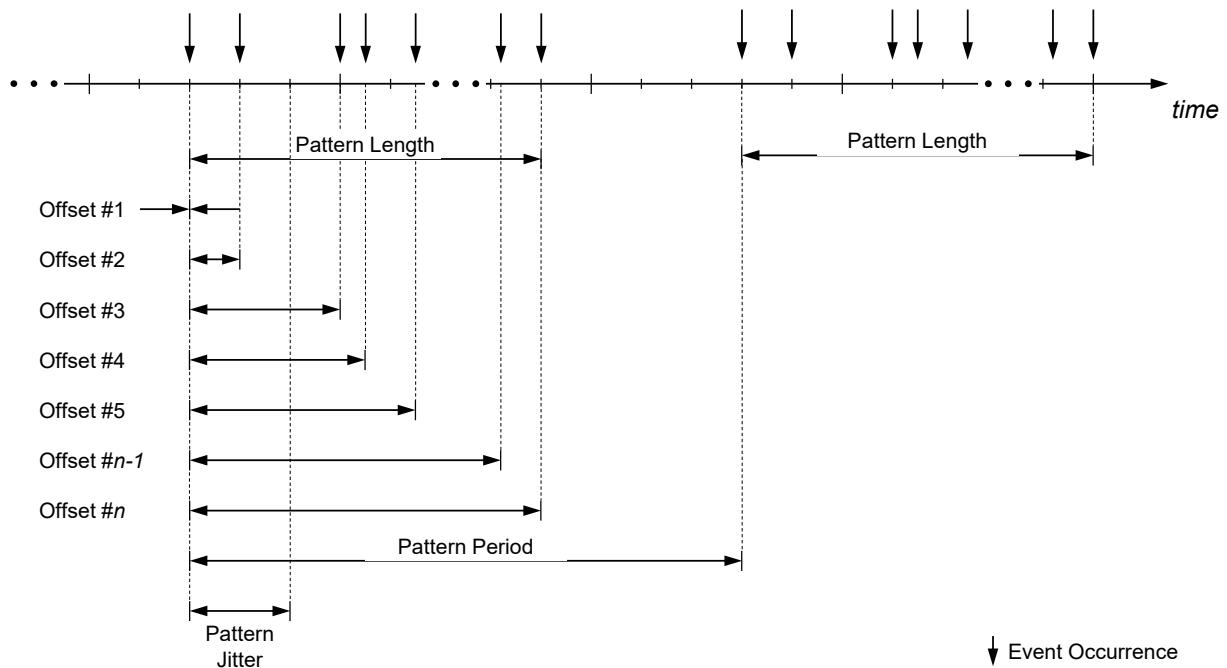


Figure 2.49: Parameters characterizing the `ConcretePatternEventTriggering` Constraint when periodically repeated

[constr_4519] Specifying `patternLength`

Imposition time: `IT_SubClasTdEvAss`

〔The `patternLength` shall be specified such that the following holds: $0 \leq \max(\text{offset}) \leq \text{patternLength}$.〕

[constr_4544] Specifying `patternLength`, `patternJitter` and `patternPeriod`

Imposition time: `IT_SubClasTdEvAss`

〔The `patternLength`, `patternJitter` and `patternPeriod` shall be specified such that the following holds: `patternLength + patternJitter < patternPeriod`.〕

[constr_6824] Existence of `ConcretePatternEventTriggering.patternLength`

Imposition time: `IT_SubClasTdEvAss`

〔For each `ConcretePatternEventTriggering`, the attribute `patternLength` shall exist.〕

[constr_6898] Existence of `ConcretePatternEventTriggering.offset`

Imposition time: `IT_SubClasTdEvAss`

〔For each `ConcretePatternEventTriggering`, the attribute `offset` shall exist.〕

2.4.1.2.4 BurstPatternEventTriggering

[TPS_TIMEX_00013] **BurstPatternEventTriggering** specifies **burst of occurrences of events**

Upstream requirements: [RS_TIMEX_00001](#), [RS_TIMEX_00002](#), [RS_TIMEX_00006](#), [RS_TIMEX_00008](#)

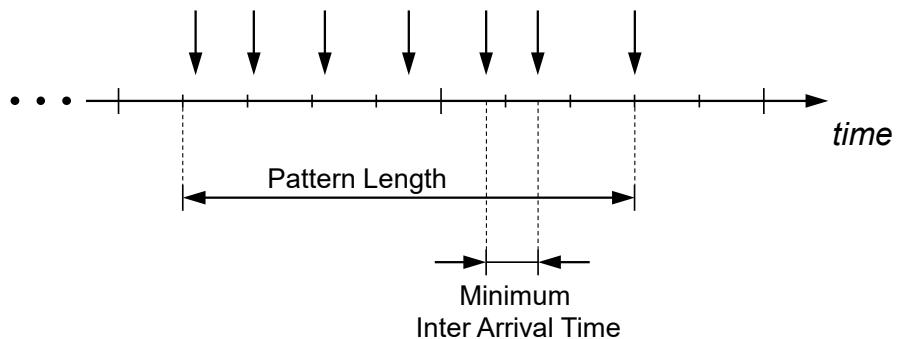
[The element [BurstPatternEventTriggering](#) is used to specify the characteristics of a burst of occurrences of one and the same [TimingDescriptionEvent](#).]

| | | | | |
|-------------------------|--|--------------|-------------|---|
| Class | BurstPatternEventTriggering | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint::EventTriggeringConstraint | | | |
| Note | Describes the maximum number of occurrences of the same event in a given time interval. Typically used to model a worst case activation scenario. | | | |
| Base | ARObject , EventTriggeringConstraint , Identifiable , MultilanguageReferrable , Referrable , Timing Constraint , Traceable | | | |
| Aggregated by | TimingExtension.timingGuarantee , TimingExtension.timingRequirement | | | |
| Attribute | Type | Mult. | Kind | Note |
| maxNumberOfOccurrences | PositiveInteger | 0..1 | attr | The maximum number of event occurrences within the given time interval. The event may never occur, or may occur N times between 1 and maxNumberOfOccurrences . If the parameter minNumberOfOccurrences is specified then the event occurs at least the number of times specified by minNumberOfOccurrences and at maximum by maxNumberOfOccurrences . |
| minimumInterArrivalTime | MultidimensionalTime | 0..1 | aggr | Specifies the minimum distance between subsequent occurrences of the event within the given time interval. |
| minNumberOfOccurrences | PositiveInteger | 0..1 | attr | The minimum number of event occurrences within the given time interval. Tags: <code>xml.sequenceOffset=10</code> |
| patternJitter | MultidimensionalTime | 0..1 | aggr | The maximum deviation of the time interval's starting point from the beginning of the given period. This parameter is only applicable in conjunction with the parameter patternPeriod |
| patternLength | MultidimensionalTime | 0..1 | aggr | The duration of the time interval within which the event repeatedly occurs. The event occurs at arbitrary points in time within the given time interval. |
| patternPeriod | MultidimensionalTime | 0..1 | aggr | The time distance between the beginnings of subsequent repetitions of the given burst pattern. |

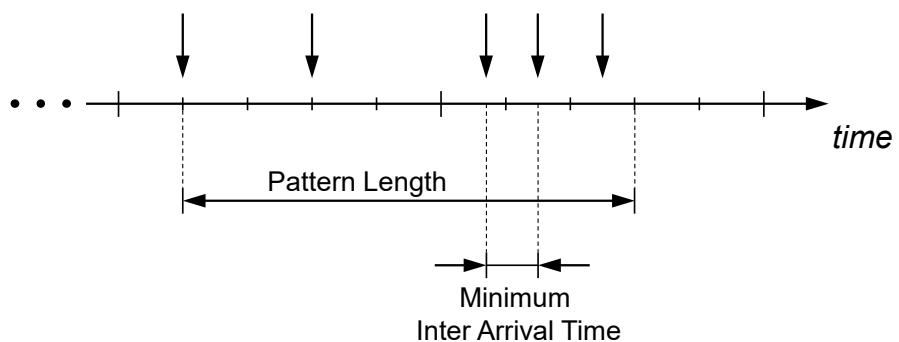
Table 2.57: BurstPatternEventTriggering

Figure 2.50 and Figure 2.51 illustrate the parameters of the [BurstPatternEventTriggering](#) in use.

Maximum Number of Occurrences = 7



Minimum Number of Occurrences = 5 (optional)

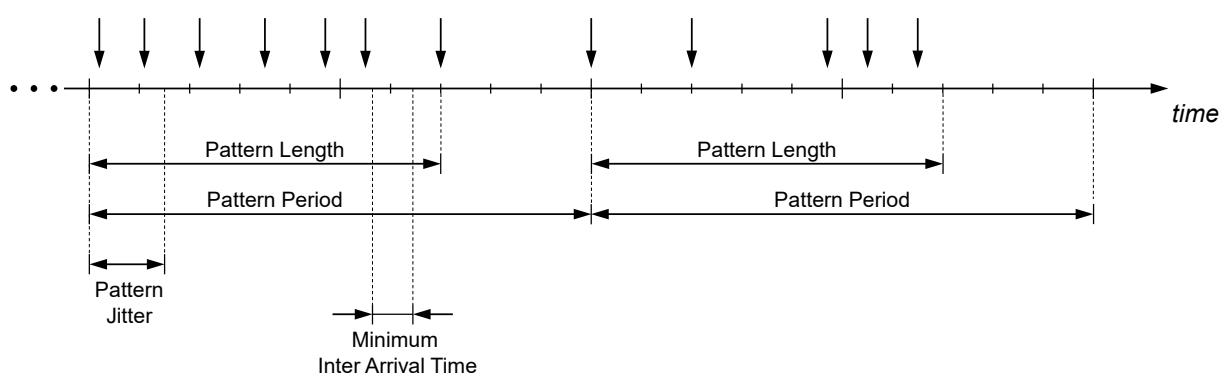


↓ Event Occurrence

Figure 2.50: Parameters characterizing the `BurstPatternEventTriggering` Constraint

Maximum Number of Occurrences = 7

Minimum Number of Occurrences = 5



↓ Event Occurrence

Figure 2.51: Parameters characterizing the `BurstPatternEventTriggering` Constraint when periodically repeated

[constr_4505] Specifying minimum and maximum number of occurrences*Imposition time:* `IT_SubClasTdEvAss`

〔The minimum and maximum number of occurrences shall be specified such that the following holds: $0 \leq \text{minNumberOfOccurrences} \leq \text{maxNumberOfOccurrences}$.〕

[constr_4506] Specifying minimum inter-arrival time and pattern length*Imposition time:* `IT_SubClasTdEvAss`

〔The `minimumInterArrivalTime` and `patternLength` shall be specified such that the following holds: $0 < \text{minimumInterArrivalTime} \leq \text{patternLength}$.〕

[constr_4507] Specifying pattern length, pattern jitter and pattern period*Imposition time:* `IT_SubClasTdEvAss`

〔The `patternLength`, `patternJitter` and `patternPeriod` shall be specified such that the following holds: `patternLength + patternJitter < patternPeriod`.〕

[constr_6825] Existence of `BurstPatternEventTriggering.maxNumberOfOccurrences`*Imposition time:* `IT_SubClasTdEvAss`

〔For each `BurstPatternEventTriggering`, the attribute `maxNumberOfOccurrences` shall exist〕

[constr_6826] Existence of `BurstPatternEventTriggering.minimumInterArrivalTime`*Imposition time:* `IT_SubClasTdEvAss`

〔For each `BurstPatternEventTriggering`, the attribute `minimumInterArrivalTime` shall exist〕

[constr_6827] Existence of `BurstPatternEventTriggering.patternLength`*Imposition time:* `IT_SubClasTdEvAss`

〔For each `BurstPatternEventTriggering`, the attribute `patternLength` shall exist〕

2.4.1.2.5 ArbitraryEventTriggering

In contrast to the `ConcretePatternEventTriggering`, this event triggering is not as strict to the occurrence of an event, but generally describes event occurrences.

[TPS_TIMEX_00014] [ArbitraryEventTriggering](#) specifies arbitrary occurrences of an event

Upstream requirements: [RS_TIMEX_00001](#), [RS_TIMEX_00002](#), [RS_TIMEX_00006](#), [RS_TIMEX_00008](#)

〔The element [ArbitraryEventTriggering](#) is used to specify the characteristics of a [TimingDescriptionEvent](#) which occurs arbitrarily.〕

| | | | | |
|----------------------|---|--------------|-------------|---|
| Class | ArbitraryEventTriggering | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint::EventTriggeringConstraint | | | |
| Note | <p>Describes that an event occurs occasionally, singly, irregularly or randomly.</p> <p>The primary purpose of this event triggering is to abstract event occurrences captured by data acquisition tools (background debugger, trace analyzer, etc.) during system runtime.</p> | | | |
| Base | <i>ARObject</i> , EventTriggeringConstraint , Identifiable , MultilanguageReferrable , Referrable , Timing Constraint , Traceable | | | |
| Aggregated by | TimingExtension.timingGuarantee , TimingExtension.timingRequirement | | | |
| Attribute | Type | Mult. | Kind | Note |
| confidence Interval | ConfidenceInterval | * | aggr | <p>List of confidence intervals.</p> <p>Tags: xml.sequenceOffset=30</p> |
| maximum Distance | MultidimensionalTime | * | aggr | <p>The nth array element describes the maximum distance that can be observed for a sample of n+1 event occurrences.</p> <p>This is an array with an identical number of elements as for the minimumDistance.</p> <p>Tags: xml.name=TIME-VALUE xml.roleElement=true xml.sequenceOffset=20 xml.typeElement=false</p> |
| minimum Distance | MultidimensionalTime | * | aggr | <p>The nth array element describes the minimum distance that can be observed for a sample of n+1 event occurrences.</p> <p>This is an array with an identical number of elements as for the maximumDistance.</p> <p>Tags: xml.name=TIME-VALUE xml.roleElement=true xml.sequenceOffset=10 xml.typeElement=false</p> |

Table 2.58: [ArbitraryEventTriggering](#)

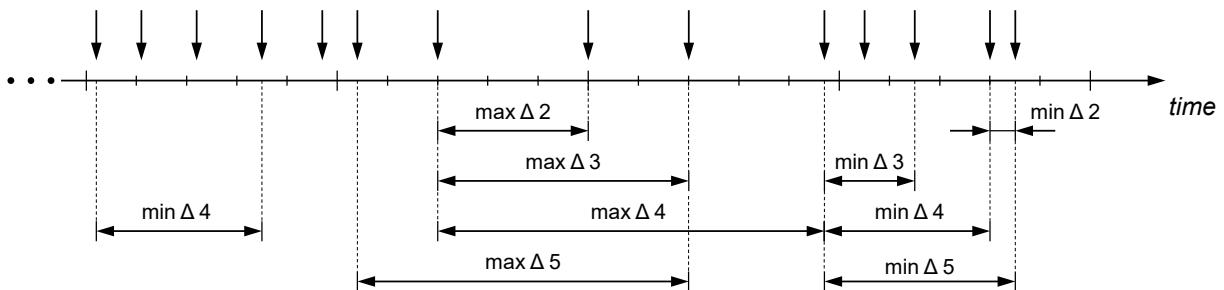
| | | | | |
|----------------------|--|--------------|-------------|--|
| Class | ConfidenceInterval | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint::EventTriggeringConstraint | | | |
| Note | Additionally to the list of measured distances of event occurrences, a confidence interval can be specified for the expected distance of two consecutive event occurrences with a given probability. | | | |
| Base | <i>ARObject</i> | | | |
| Aggregated by | ArbitraryEventTriggering.confidenceInterval | | | |
| Attribute | Type | Mult. | Kind | Note |
| lowerBound | MultidimensionalTime | 0..1 | aggr | The lower bound of the expected distance of two consecutive event occurrences. |



| Class | ConfidenceInterval | | | |
|-------------|----------------------|------|-------|--|
| propability | Float | 0..1 | attr | The probability for the measured lower and upper bound of the confidence interval. |
| upperBound | MultidimensionalTime | 0..1 | agrgr | The upper bound of the expected distance of two consecutive event occurrences. |

Table 2.59: ConfidenceInterval

Figure 2.52 illustrates the parameters of the [ArbitraryEventTriggering](#) in use.



$\min \Delta n$ Least minimum inter-arrival time between n subsequent occurrences of the event E and $n = \{2, 3, 4, \dots\}$

$\max \Delta n$ Major maximum inter-arrival time between n subsequent occurrences of the event E and $n = \{2, 3, 4, \dots\}$

↓ Event Occurrence

Figure 2.52: Parameters characterizing the ArbitraryEventTriggering Constraint

[constr_6828] Existence of [ArbitraryEventTriggering.minimumDistance](#)

Imposition time: [IT_SubClasTdEvAss](#)

For each [ArbitraryEventTriggering](#), the reference in the role [minimumDistance](#) shall exist at least once]

[constr_6829] Existence of [ArbitraryEventTriggering.maximumDistance](#)

Imposition time: [IT_SubClasTdEvAss](#)

For each [ArbitraryEventTriggering](#), the reference in the role [maximumDistance](#) shall exist at least once]

[constr_6830] Existence of [ConfidenceInterval.lowerBound](#)

Imposition time: [IT_SubClasTdEvAss](#)

For each [ConfidenceInterval](#), the attribute [lowerBound](#) shall exist]

[constr_6831] Existence of [ConfidenceInterval.propability](#)

Imposition time: [IT_SubClasTdEvAss](#)

For each [ConfidenceInterval](#), the attribute [propability](#) shall exist]

[constr_6832] Existence of [ConfidenceInterval.upperBound](#)

Imposition time: [IT_SubClasTdEvAss](#)

〔For each [ConfidenceInterval](#), the attribute [upperBound](#) shall exist〕

2.4.1.3 [OffsetTimingConstraint](#)

An [OffsetTimingConstraint](#) bounds the time offset between the occurrence of two timing events, without requiring a direct functional dependency between the source and the target.

This constraint type is frequently used in combination with the timing event [TDE-ventCycleStart](#) as source. In this case, the target event (e.g. the start of a [RunnableEntity](#)) is in most of the cases functional independent from the the source event.

[TPS_TIMEX_00015] [OffsetTimingConstraint](#) specifies offset between occurrences of events

Upstream requirements: [RS_TIMEX_00001](#), [RS_TIMEX_00002](#), [RS_TIMEX_00008](#)

〔The element [OffsetTimingConstraint](#) is used to specify an offset between the occurrences of two [TimingDescriptionEvent](#)s.〕

| Class | OffsetTimingConstraint | | | |
|---------------|--|-------|------|---|
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint::OffsetConstraint | | | |
| Note | Bounds the time offset between the occurrence of two timing events, without requiring a direct functional dependency between the source and the target . If the target event occurs, it is expected to occur earliest with the minimum , and latest with the maximum offset relatively after the occurrence of the source event. Note: not every source event occurrence shall be followed by a target event occurrence. In contrast to LatencyTimingConstraint , there shall not necessarily be a causal dependency between the source and target event. | | | |
| Base | ARObject , Identifiable , MultilanguageReferrable , Referrable , TimingConstraint , Traceable | | | |
| Aggregated by | TimingExtension.timingGuarantee , TimingExtension.timingRequirement | | | |
| Attribute | Type | Mult. | Kind | Note |
| maximum | MultidimensionalTime | 0..1 | aggr | The maximum offset the target event occurs relatively after the occurrence of the source event. Tags: xml.sequenceOffset=20 |
| minimum | MultidimensionalTime | 0..1 | aggr | The minimum offset the target event occurs relatively after the occurrence of the source event. Tags: xml.sequenceOffset=10 |
| source | TimingDescriptionEvent | 0..1 | ref | The timing event that the target event is to be synchronized with. |
| target | TimingDescriptionEvent | 0..1 | ref | The timing event which is expected to occur timely after the source event. |

Table 2.60: [OffsetTimingConstraint](#)

[constr_6842] Existence of `OffsetTimingConstraint.maximum`

Imposition time: `IT_SubClasTdEv`

〔For each `OffsetTimingConstraint`, the attribute `maximum` shall exist〕

[constr_6843] Existence of `OffsetTimingConstraint.minimum`

Imposition time: `IT_SubClasTdEv`

〔For each `OffsetTimingConstraint`, the attribute `minimum` shall exist〕

[constr_6844] Existence of `OffsetTimingConstraint.source`

Imposition time: `IT_SubClasTdEv`

〔For each `OffsetTimingConstraint`, the reference in the role `source` shall exist at least once〕

[constr_6845] Existence of `OffsetTimingConstraint.target`

Imposition time: `IT_SubClasTdEv`

〔For each `OffsetTimingConstraint`, the reference in the role `target` shall exist at least once〕

2.4.1.4 AgeConstraint

Sometimes it is necessary to specify the age of data, when it arrives at a SWC on its required port with `SenderReceiverInterface`. If the sender of the data is known, a `TimingDescriptionEventChain` can be defined from the sender to the receiver port and a `LatencyTimingConstraint` with age semantics represents the specification of the data age. However, the actual sender of the data may not be known, in which case the definition of a `TimingDescriptionEventChain` is not possible.

Typically, the `AgeConstraint` restricts the time interval between the physical creation of data by the corresponding sender and the availability of the data in the receiver. The scope of an `AgeConstraint` is a `TDEventVariableDataPrototype`. Every time the scoped event occurs, the `VariableDataPrototype` shall have the specified data age.

At a later stage during the development, when the refined software architecture exposes the relation between the actual sender of the data and the receiver, an event chain between the sending and receiving point in time shall be defined and associated with a `LatencyTimingConstraint` (see 2.4.2.2) in order to refine the previous defined age constraint.

| Class | AgeConstraint | | | |
|---------------|---|-------|------|--|
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint::AgeConstraint | | | |
| Note | Constrains the <code>scope</code> by a <code>minimum</code> and <code>maximum</code> time boundary. | | | |
| Base | <code>ARObject</code> , <code>Identifiable</code> , <code>MultilanguageReferrable</code> , <code>Referrable</code> , <code>TimingConstraint</code> , <code>Traceable</code> | | | |
| Aggregated by | <code>TimingExtension.timingGuarantee</code> , <code>TimingExtension.timingRequirement</code> | | | |
| Attribute | Type | Mult. | Kind | Note |
| maximum | MultidimensionalTime | 0..1 | aggr | The received event referenced by <code>scope</code> should not exceed this upper bound. |
| minimum | MultidimensionalTime | 0..1 | aggr | The received event referenced by <code>scope</code> should not precede this lower bound. |
| scope | <code>TimingDescriptionEvent</code> | 0..1 | ref | <code>TimingDescriptionEvent</code> to be constrained. |

Table 2.61: AgeConstraint

[constr_4504] Restriction of the `scope` of an AgeConstraint

Imposition time: `IT_SubClasTdEvAss`

[An `AgeConstraint` may only reference either a:

- `TDEventVariableDataPrototype.tdEventVariableDataPrototype-`
`Type==variableDataPrototypeReceived`
- `TDEventTrigger.tdEventTriggerType==triggerActivated`

in the role `scope`]

Since an `AgeConstraint` is intended to be used only in those scenarios without a sending context it is prudent to restrict it to the receiving side context of the intended sub-class of `TimingDescriptionEvent` and further constrain the referenced port context to receiver side `PortPrototype`s.

[constr_6914] Restriction of the `port` context of an AgeConstraint

Status: DRAFT

Imposition time: `IT_SubClasTdEvAss`

[An `AgeConstraint.scope.port` shall reference only sub-classes of `AbstractRequiredPortPrototype`]

2.4.2 TimingConstraints on TimingDescriptionEventChains

A `TimingConstraint` on a `TimingDescriptionEventChain` imposes a restriction/expectation on level of the *whole* `TimingDescriptionEventChain`. In particular the `LatencyTimingConstraint` is used to specify a reaction and age, for example if a stimulus event occurs then the corresponding response event shall occur not later than a given amount of time. The `SynchronizationTimingConstraint`

is used to specify that stimuli or response events shall occur within a given time interval (tolerance) to be said to occur simultaneous and synchronous respectively.

Generally a `TimingConstraint`, when applied to `TimingDescriptionEventChain`, refers to a given scope/chain context with a temporal restriction, present in most sub-classes. The remaining attributes are context specific. As the `TimingDescriptionEventChain` has a semantic of a directed acyclic graph, the direction is obvious, but it matters whether a single `TimingDescriptionEventChain` or a group of `TimingDescriptionEventChain` are constrained.

In 2.53 differing sub-classes of `TimingConstraints` on `TimingDescription-EventChains` are shown.

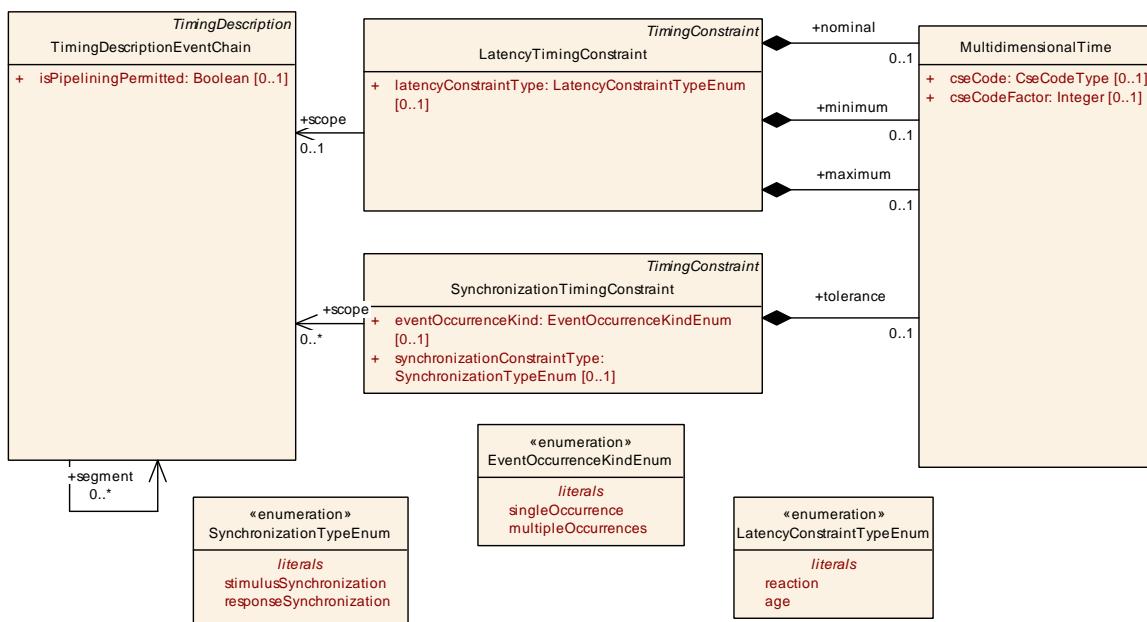


Figure 2.53: TimingConstraints on TimingDescriptionEventChains

2.4.2.1 SynchronizationTimingConstraint

The objective of synchronization in a distributed environment is to establish and maintain a consistent time base for the interaction between different subsystems, in order to obtain correct runtime order and avoid unexpected race conditions. While mechanisms to establish synchronization need to be provided at the implementation level, the necessity for synchronization needs to be expressed at design level. For this purpose, synchronization constraints are used.

A `SynchronizationTimingConstraint` imposes a synchronization restriction among either the `stimulus` or `response` event occurrences of two or more `TimingDescriptionEventChains`. In the former case (stimulus synchronization) the referenced `TimingDescriptionEventChains` shall have the same `response` event (join), or in the latter case (response synchronization) they shall have the same `stimulus` event (fork).

| | | | | |
|--------------------------------|--|--------------|-------------|---|
| Class | SynchronizationTimingConstraint | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint::SynchronizationTiming Constraint | | | |
| Note | <p>This constraint is used to restrict the timing behavior of different, but correlated events or event chains, with regard to synchronization. Two scenarios are supported:</p> <ul style="list-style-type: none"> • If (<code>synchronizationConstraintType==responseSynchronization</code>) <ul style="list-style-type: none"> – <code>TimingDescriptionEvent</code>s: An arbitrary number of correlated events which play the role of responses shall occur synchronously with respect to a predefined tolerance. – <code>TimingDescriptionEventChains</code>: An arbitrary number of correlated event chains with a common stimulus, but different responses, where the responses shall occur synchronously with respect to a predefined tolerance. • If (<code>synchronizationConstraintType==stimulusSynchronization</code>) <ul style="list-style-type: none"> – <code>TimingDescriptionEvent</code>s: An arbitrary number of correlated events which play the role of stimuli shall occur synchronously with respect to a predefined tolerance. – <code>TimingDescriptionEventChains</code>: An arbitrary number of correlated event chains with a common response, but different stimuli, where the stimuli shall occur synchronously with respect to a predefined tolerance. <p>In case the constraint is imposed on events the following two scenarios are supported:</p> <ul style="list-style-type: none"> • If (<code>eventOccurrenceKind==singleOccurrence</code>): any of the events shall occur only once in the given time interval. • If (<code>eventOccurrenceKind==multipleOccurrences</code>): any of the events may occur more than once in the given time interval. In other words multiple occurrences of an event within the given time interval are permitted. | | | |
| Base | <code>ARObject</code> , <code>Identifiable</code> , <code>MultilanguageReferrable</code> , <code>Referrable</code> , <code>TimingConstraint</code> , <code>Traceable</code> | | | |
| Aggregated by | <code>TimingExtension.timingGuaranteee</code> , <code>TimingExtension.timingRequirement</code> | | | |
| Attribute | Type | Mult. | Kind | Note |
| event OccurrenceKind | <code>EventOccurrenceKind</code> <code>Enum</code> | 0..1 | attr | Indicates whether the referenced events shall occur only once (single occurrence) or multiple times (multiple occurrences) in the given time interval. |
| scope | <code>TimingDescriptionEvent</code> <code>Chain</code> | * | ref | The event chains that are in the scope of the constraint. Mutually exclusive to <code>scopeEvent</code> , see ([<code>constr_4522</code>]). |
| scopeEvent | <code>TimingDescriptionEvent</code> | * | ref | The events that are in the scope of the constraint. Mutually exclusive to <code>scope</code> , see ([<code>constr_4522</code>]). |
| synchronization ConstraintType | <code>SynchronizationType</code> <code>Enum</code> | 0..1 | attr | Indicates whether the associated events of the <code>SynchronizationTimingConstraint</code> have a common stimulus or response. |
| tolerance | <code>MultidimensionalTime</code> | 0..1 | aggr | The maximum time interval, within which the synchronized events shall occur. The events may occur in any order within this time interval. The time interval starts at the point-in-time when one of the referenced events occurs. |

 Table 2.62: **SynchronizationTimingConstraint**

| | |
|----------------------|---|
| Enumeration | SynchronizationTypeEnum |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint::SynchronizationTiming Constraint |
| Note | Specifies the <code>synchronizationConstraintType</code> for a <code>SynchronizationTimingConstraint</code> . |
| Aggregated by | <code>SynchronizationTimingConstraint.synchronizationConstraintType</code> |
| Literal | Description |



△

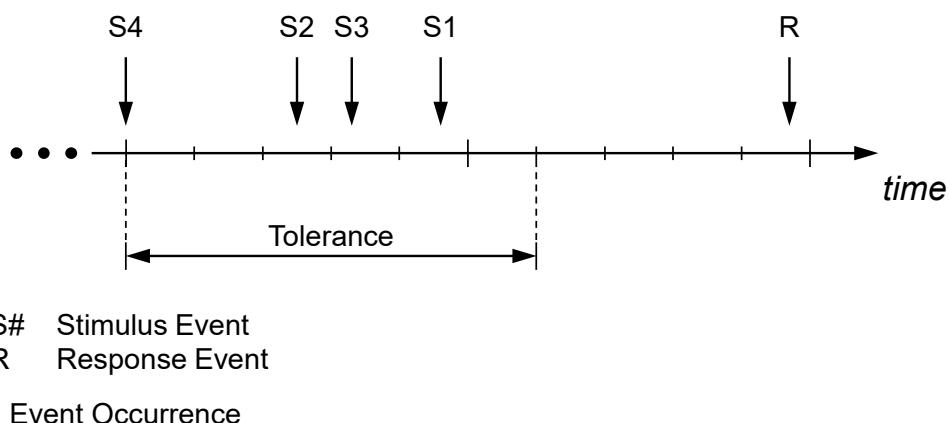
| Enumeration | SynchronizationTypeEnum |
|--------------------------|--|
| response Synchronization | <p>In case that the Synchronization Timing Constraint is specified for event chains, the response events of the associated event chains shall occur synchronously with respect to the specified tolerance. All associated event chains shall have the same stimulus event.</p> <p>In case that the Synchronization Timing Constraint is specified for events, the associated events shall occur synchronously with respect to the specified tolerance. All associated events represent the response events of a common stimulus event, even such a stimulus event is not known yet or not available in the scope of the model.</p> <p>Tags: atp.EnumerationLiteralIndex=0</p> |
| stimulus Synchronization | <p>In case that the Synchronization Timing Constraint is specified for event chains, the stimulus events of the associated event chains shall occur synchronously with respect to the specified tolerance. All associated event chains shall have the same response event.</p> <p>In case that the Synchronization Timing Constraint is specified for events, the associated events shall occur synchronously with respect to the specified tolerance. All associated events represent the stimulus events of a common response event, even such a response event is not known yet or not available in the scope of the model.</p> <p>Tags: atp.EnumerationLiteralIndex=1</p> |

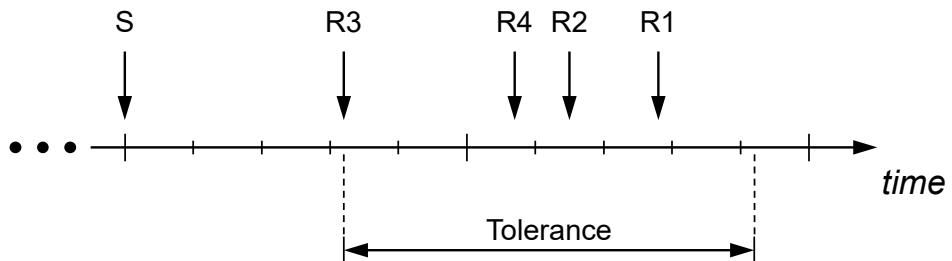
Table 2.63: SynchronizationTypeEnum

| Enumeration | EventOccurrenceKindEnum |
|----------------------|--|
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint::SynchronizationTiming Constraint |
| Note | Specifies the eventOccurrenceKind for a SynchronizationTimingConstraint . |
| Aggregated by | SynchronizationTimingConstraint.eventOccurrenceKind |
| Literal | Description |
| multiple Occurrences | Specifies that an event may occur more than once in a given time interval. Tags: atp.EnumerationLiteralIndex=0 |
| singleOccurrence | The referenced event shall occur only once in a given time interval. Indicates whether the referenced events shall occur only once (single occurrence) or multiple times (multiple occurrences) in the given time interval. Tags: atp.EnumerationLiteralIndex=1 |

Table 2.64: EventOccurrenceKindEnum

The parameters are shown in use in Figure 2.54 and Figure 2.55.


 Figure 2.54: [SynchronizationTimingConstraint](#) imposed on [TimingDescription-EventChain.stimulus](#)



S Stimulus Event
 R# Response Event
 ↓ Event Occurrence

Figure 2.55: SynchronizationTimingConstraint imposed on TimingDescriptionEventChain.response

An example for synchronizing on *stimuli* of *TimingDescriptionEventChains* would be an adaptive cruise control that expects data from different sensors, which shall be sampled (quasi) simultaneously with respect to a predefined tolerance.

An example for synchronizing on *responses* of *TimingDescriptionEventChains* would be the blinking of different indicator lights, which shall occur (quasi) simultaneously with respect to a predefined tolerance.

[constr_4522] SynchronizationTimingConstraint shall either reference events or event chains

Imposition time: IT_SubClasTdEvAss

[The *SynchronizationTimingConstraint* shall either reference *TimingDescriptionEvents* or *TimingDescriptionEventChains*, but not both at the same time.]

[constr_4514] SynchronizationTimingConstraint shall reference at least two event chains

Imposition time: IT_SubClasTdEvAss

[In the case, that the *SynchronizationTimingConstraint* is imposed on *TimingDescriptionEventChains* then at least two (2) *TimingDescriptionEventChains* shall be referenced.]

[constr_4521] Specifying attribute synchronizationConstraintType

Imposition time: IT_SubClasTdEvAss

[The attribute *synchronizationConstraintType* shall be specified if the *SynchronizationTimingConstraint* is imposed on *TimingDescriptionEventChains*.]

2.4.2.2 LatencyTimingConstraint

A [LatencyTimingConstraint](#) specifies a [minimum](#) and/or [maximum](#) time duration between the occurrence of the [stimulus](#) and the occurrence of the corresponding [response](#) of that [TimingDescriptionEventChain](#). For example, this could be the time it takes for a packet of data on a bus network to get from one designated point to another, or the time it takes for a function/task to be executed on a processor.

[TPS_TIMEX_00004] [LatencyTimingConstraint](#) specifies latency constraints

Upstream requirements: [RS_TIMEX_00001](#), [RS_TIMEX_00002](#), [RS_TIMEX_00012](#), [RS_TIMEX_00015](#)

[The element [LatencyTimingConstraint](#)² is used to specify the amount of time that elapses between the occurrence of any two [TimingDescriptionEvents](#).]

| Class | LatencyTimingConstraint | | | |
|-----------------------|--|-------|------|--|
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint::LatencyTimingConstraint | | | |
| Note | Constrains the time duration between the occurrence of the stimulus and the occurrence of the corresponding response of that scope . In contrast to scope , a causal dependency between the stimulus and the corresponding response of the scope is required. | | | |
| Base | ARObject , Identifiable , MultilanguageReferrable , Referrable , TimingConstraint , Traceable | | | |
| Aggregated by | TimingExtension.timingGuarantee , TimingExtension.timingRequirement | | | |
| Attribute | Type | Mult. | Kind | Note |
| latencyConstraintType | LatencyConstraintType Enum | 0..1 | attr | The specific type of this latency constraint. |
| maximum | MultidimensionalTime | 0..1 | aggr | The maximum latency between the occurrence of the stimulus and the occurrence of the corresponding response of the associated event chain. Tags: xml.sequenceOffset=20 |
| minimum | MultidimensionalTime | 0..1 | aggr | The minimum latency between the occurrence of the stimulus and the occurrence of the corresponding response of the associated event chain. Tags: xml.sequenceOffset=10 |
| nominal | MultidimensionalTime | 0..1 | aggr | The nominal latency between the occurrence of the stimulus and the occurrence of the corresponding response of the associated event chain. Tags: xml.sequenceOffset=30 |
| scope | TimingDescriptionEvent Chain | 0..1 | ref | The event chain that defines the scope of the constraint. |

Table 2.65: [LatencyTimingConstraint](#)

²A synonym for delay

| Enumeration | LatencyConstraintTypeEnum |
|---------------|--|
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint::LatencyTimingConstraint |
| Note | Specifies the <code>latencyConstraintType</code> for a <code>LatencyTimingConstraint</code> . |
| Aggregated by | <code>LatencyTimingConstraint.latencyConstraintType</code> |
| Literal | Description |
| age | The <code>LatencyTimingConstraint</code> is seen from the perspective of the <code>response</code> event of the <code>scope</code> . Given a certain <code>response</code> event, the age interval of the latest <code>stimulus</code> is constrained. Tags: atp.EnumerationLiteralIndex=0 |
| reaction | The <code>LatencyTimingConstraint</code> is seen from the perspective of the <code>stimulus</code> event of the <code>scope</code> . Given a certain <code>stimulus</code> event, the reaction interval of the first <code>response</code> is constrained. Tags: atp.EnumerationLiteralIndex=1 |

Table 2.66: LatencyConstraintTypeEnum

The attributes `minimum`, `maximum`, and `nominal` of a `LatencyTimingConstraint` can be used to define a lower and upper bound, as well as a nominal value for the latency of the `TimingDescriptionEventChain` in the `scope`.

Example: In multi-rate networks, data can get lost or get duplicated because of potential different producer and consumer periods. Data loss occurs, if the consumer's period is greater than the producer's period (under-sampling). Accordingly, data duplication occurs, if the consumer's period is smaller than the producer's period (oversampling). This is depicted in figure 2.56.

The application of `LatencyTimingConstraints` leads to some interesting observations:

- In systems without over- and under-sampling, `age` and `reaction` are the same. But timing constraints are implementation-independent. Thus, at specification time when the implementation is not necessarily known, the correct latency constraint semantics has to be specified.
- The minimum reaction and the minimum age latency of an event chain are always equal.

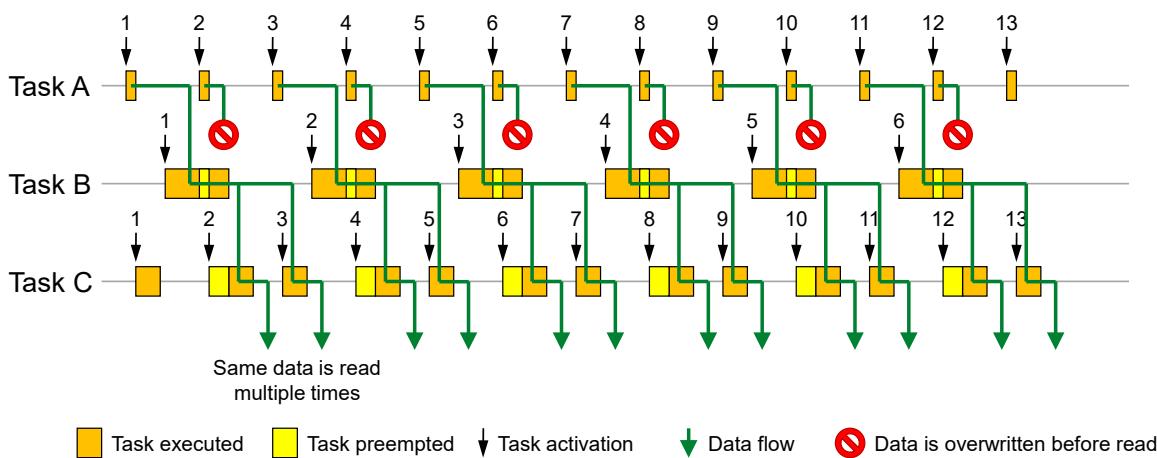


Figure 2.56: Loss and duplication of data due to under- and oversampling.

Considering under- and oversampling, two end-to-end latency semantics are of interest for automotive systems and can thus be expressed with TIMEX. These are the *age* of a certain response and the *reaction* to a certain stimulus.

The *data age timing constraint* is mainly important in control engineering, but may appear in all domains. Here the focus is from the response perspective rather than from the stimulus perspective. In other words, the assumption is that last is best, i.e., it is accepted/tolerated that a value is overwritten along the path from stimulus to response. When for example an actuator value is periodically updated, it is of importance that the corresponding input values are not too old. In this case the constrained time of importance is the delay from the latest stimulus to a given response.

The *reaction time constraint* is utilized when the first reaction to a stimulus is of importance. This is usually the case in body electronics, but may also be the case in other domains. One example is the time it takes from a button is pressed to the light is switched on. Another example, from the chassis domain, is the time from the brake pedal is pressed until the brakes are activated. In both cases the constrained time of importance is the delay from a given stimulus to the first corresponding response.

[constr_6837] Existence of `LatencyTimingConstraint.latencyConstraintType`

Imposition time: `IT_SubClasTdEvAss`

〔For each `LatencyTimingConstraint`, the attribute `latencyConstraintType` shall exist〕

[constr_6838] Existence of `LatencyTimingConstraint.maximum`

Imposition time: `IT_SubClasTdEvAss`

〔For each `LatencyTimingConstraint`, the attribute `maximum` shall exist〕

[constr_6839] Existence of `LatencyTimingConstraint.minimum`

Imposition time: `IT_SubClasTdEvAss`

〔For each `LatencyTimingConstraint`, the attribute `minimum` shall exist〕

[constr_6841] Existence of `LatencyTimingConstraint.scope`

Imposition time: `IT_SubClasTdEvAss`

〔For each `LatencyTimingConstraint`, the reference in the role `scope` shall exist at least once〕

2.4.3 TimingConstraints on ExecutableEntities

TIMEX provides the means to specify constraints on [ExecutableEntity](#)s around execution order or an execution time etc. In contrast to the constraints on single events or event chains, the approach is a little different and is tied together with the use-cases e.g. [3.1](#), where several types of constraints are applied depending on the configuration. In this chapter, the general modeling is explained.

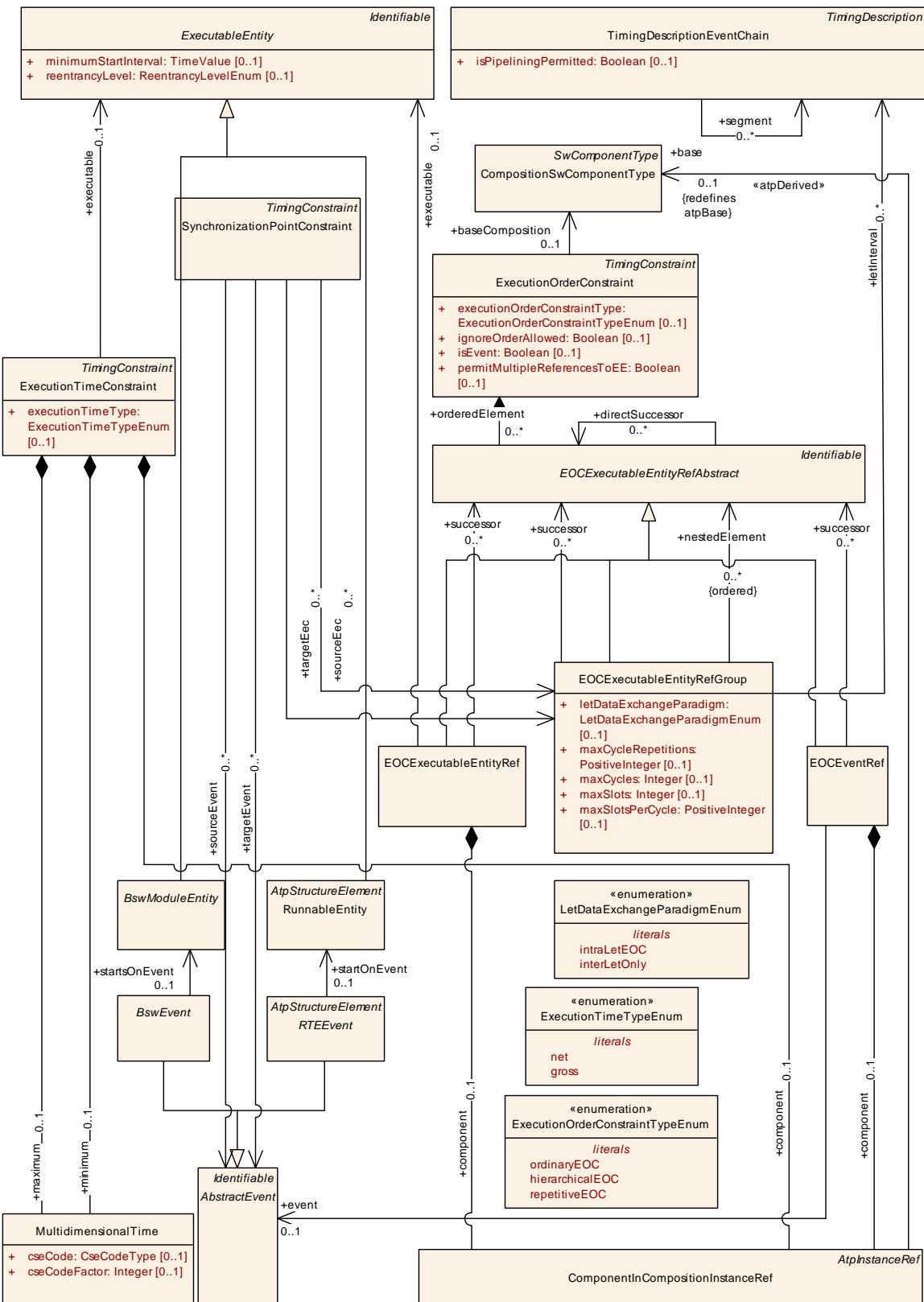


Figure 2.57: TimingConstraints on ExecutableEntities

2.4.3.1 ExecutionOrderConstraint

An [ExecutionOrderConstraint](#)³ can be used in any TIMEX view, as long as the [ExecutableEntity](#)s and/or the [AbstractEvent](#)s to be referenced are available in other AUTOSAR models, namely the [6] and [9].

[TPS_TIMEX_00007] [ExecutionOrderConstraint](#) constrains the ordering of sequences of [ExecutableEntity](#)s

Upstream requirements: [RS_TIMEX_00001](#), [RS_TIMEX_00002](#), [RS_TIMEX_00014](#)

[The element [ExecutionOrderConstraint](#) is used to specify the order of execution of a number of [ExecutableEntity](#)s.]

| | | | | |
|-------------------------------|--|--------------|-------------|---|
| Class | ExecutionOrderConstraint | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint::ExecutionOrderConstraint | | | |
| Note | <p>This constraint is used to restrict the order of execution for a set of ExecutableEntitys. The ExecutionOrderConstraint can be used in any timing view.</p> <p>The various scopes for ExecutionOrderConstraint are described below. Generally, each ExecutionOrderConstraint has a scope of software components and can reference all ExecutableEntitys available in the corresponding internal behavior (RunnableEntity and BswModuleEntity) either directly or by the events activating respectively starting them (RteEvent and BswEvent).</p> <p>On VFB level an ExecutionOrderConstraint can be specified for RunnableEntities part of the composition hierarchy referenced by the VfbTiming.</p> <p>On SW-C level an ExecutionOrderConstraint can be specified for RunnableEntities part of the Internal Behavior referenced by the SwcTiming.</p> <p>On System level an ExecutionOrderConstraint can be specified for RunnableEntities part of the composition hierarchy of the system referenced by the SystemTiming.</p> <p>On BSW Module level, an ExecutionOrderConstraint can be specified for BswModuleEntities part of an BswInternalBehavior referenced by the BswModuleTiming.</p> <p>On ECU level an ExecutionOrderConstraint can be specified for all ExecutableEntitys and Events available via the EcucValueCollection, covering ECU Extract and BSW Module Configuration, referenced by the EcuTiming.</p> | | | |
| Base | ARObject , Identifiable , MultilanguageReferrable , Referrable , TimingConstraint , Traceable | | | |
| Aggregated by | TimingExtension.timingGuarantee , TimingExtension.timingRequirement | | | |
| Attribute | Type | Mult. | Kind | Note |
| base Composition | CompositionSwComponentType | 0..1 | ref | Specifies the composition SW-C type playing the role of a SW-C containing further SW-Cs and represents the scope of the Execution Order Constraint. |
| executionOrder ConstraintType | ExecutionOrderConstraintTypeEnum | 0..1 | attr | Specifies the specific type of ExecutionOrderConstraint . |
| ignoreOrder Allowed | Boolean | 0..1 | attr | Controls whether the order of execution specified by this constraint can be intentionally ignored (TRUE), or shall be respected (FALSE). |
| | | | | Tags: atp.Status=obsolete |



³Note: At first glance there may be some similarity between an [ExecutionOrderConstraint](#) and "Affinity Constraints" in [3], however the focus of [ExecutionOrderConstraint](#) is to provide a more detailed focus on the timing aspects and less any actual mapping affinity



| Class | ExecutionOrderConstraint | | | |
|------------------------------|--------------------------------|------|------|---|
| isEvent | Boolean | 0..1 | attr | Indicates whether the ExecutionOrderConstraint is only referring to Executable Entities (FALSE) or only to RTE and/or BSW Events (TRUE). |
| orderedElement | EOCExecutableEntityRefAbstract | * | aggr | This aggregation represents an unordered collection of references to RunnableEntities which shall be considered in the ExecutionOrderConstraint. The role does not imply that the collection of references itself shall be ordered. |
| permitMultipleReferencesToEE | Boolean | 0..1 | attr | Indicates that the ExecutionOrderConstraints permits that an Executable Entity is referenced multiple times (TRUE) or only once (FALSE) in the constraint. |

Table 2.67: ExecutionOrderConstraint

| Enumeration | ExecutionOrderConstraintTypeEnum |
|-----------------|--|
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint::ExecutionOrderConstraint |
| Note | Specifies the type of the executionOrderConstraintType for a ExecutionOrderConstraint . |
| Aggregated by | ExecutionOrderConstraint.executionOrderConstraintType |
| Literal | Description |
| hierarchicalEOC | Specifies that the Execution Order Constraint specifies a hierarchical execution order constraint. Tags: atp.EnumerationLiteralIndex=0 |
| ordinaryEOC | Specifies that the Execution Order Constraint specifies an ordinary execution order constraint. Tags: atp.EnumerationLiteralIndex=1 |
| repetitiveEOC | Specifies that the Execution Order Constraint specifies a repetitive execution order constraint. Tags: atp.EnumerationLiteralIndex=2 |

Table 2.68: ExecutionOrderConstraintTypeEnum

| Class | EOCExecutableEntityRefAbstract (abstract) | | | |
|-----------------|---|-------|------|---|
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint::ExecutionOrderConstraint | | | |
| Note | This is the abstractions for Execution Order Constraint Executable Entity References (leaves) and Execution Order Constraint Executable Entity Reference Groups (composites). | | | |
| Base | ARObject , Identifiable , MultilanguageReferrable , Referrable | | | |
| Subclasses | EOCEventRef , EOCExecutableEntityRef , EOCExecutableEntityRefGroup | | | |
| Aggregated by | ExecutionOrderConstraint.orderedElement | | | |
| Attribute | Type | Mult. | Kind | Note |
| directSuccessor | EOCExecutableEntityRefAbstract | * | ref | The direct successor of an executable entity or a group of executable entities. |

Table 2.69: EOCExecutableEntityRefAbstract

| Class | EOCExecutableEntityRefGroup | | | |
|---------------|--|--|--|--|
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint::ExecutionOrderConstraint | | | |
| Note | This is used to specify a group (composite) consisting of Execution Order Constraint Executable Entity References (leaves) and/or further Execution Order Constraint Executable Entity Reference Groups (composite). | | | |
| Base | ARObject , EOCExecutableEntityRefAbstract , Identifiable , MultilanguageReferrable , Referrable | | | |
| Aggregated by | ExecutionOrderConstraint.orderedElement | | | |





| Class | EOCExecutableEntityRefGroup | | | |
|---------------------------|---------------------------------|-------|------|---|
| Attribute | Type | Mult. | Kind | Note |
| letData Exchange Paradigm | LetDataExchange ParadigmEnum | 0..1 | attr | <p>Specifies the data exchange paradigm between ExecutableEntitys within a LET interval.</p> <p>Tags: atp.Status=draft</p> |
| letInterval | TimingDescriptionEvent Chain | * | ref | <p>This association references the TimingDescriptionEvent Chain that plays the role of a LET interval the executable entities in the group are assigned to.</p> <p>[constr_4554] applies.</p> |
| maxCycle Repetitions | PositiveInteger | 0..1 | attr | <p>Repetitive Execution Order Constraint only:</p> <p>The number of repetitions (cycles) of the event in the Repetitive Execution Order Constraint.</p> <p>Tags: atp.Status=draft</p> |
| maxCycles | Integer | 0..1 | attr | <p>In case of a Repetitive Execution Order Constraint this attribute specifies the number of cycles the Execution Order Constraint is considering.</p> <p>Tags: atp.Status=obsolete</p> |
| maxSlots | Integer | 0..1 | attr | <p>In case of a Repetitive Execution Order Constraint this attribute specifies the number of slots every cycle of the Execution Order Constraint is consisting of.</p> <p>Tags: atp.Status=obsolete</p> |
| maxSlotsPer Cycle | PositiveInteger | 0..1 | attr | <p>Repetitive Execution Order Constraint only:</p> <p>The number of ExecutableEntitys (slots) that are executed in a given order within a cycle, for the Repetitive Execution Order Constraint.</p> <p>Tags: atp.Status=draft</p> |
| nestedElement (ordered) | EOCExecutableEntity RefAbstract | * | ref | This association is used to establish hierarchies of EOCEER Groups and References. |
| successor | EOCExecutableEntity RefAbstract | * | ref | The logical successor of an executable entity or a group of executable entities. |
| triggeringEvent | TimingDescriptionEvent | 0..1 | ref | In case of a Repetitive Execution Order Constraint this association references the timing description event triggering every cycle. |

Table 2.70: EOCExecutableEntityRefGroup

| Class | EOCExecutableEntityRef | | | |
|--------------------|---|-------|------|---|
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint::ExecutionOrderConstraint | | | |
| Note | This is used to define a reference to an ExecutableEntity If the ExecutionOrderConstraint is defined on VFB, System or ECU level, a reference to the Sw ComponentPrototype, via the ComponentInCompositionInstanceRef, the referenced ExecutableEntity belongs to, shall be provided as context information. | | | |
| Base | ARObject, EOCExecutableEntityRefAbstract , Identifiable , MultilanguageReferrable , Referrable | | | |
| Aggregated by | ExecutionOrderConstraint.orderedElement | | | |
| Attribute | Type | Mult. | Kind | Note |
| bswModule Instance | BswImplementation | 0..1 | ref | Specifies the BSW module instance the BSW module entity belongs to. |





| Class | EOCExecutableEntityRef | | | |
|------------|---------------------------------|------|------|---|
| component | SwComponent Prototype | 0..1 | iref | <p>This association references the specific instance of the SW-C prototype.</p> <p>InstanceRef implemented by: ComponentInCompositionInstanceRef</p> |
| executable | ExecutableEntity | 0..1 | ref | The ExecutableEntity whose execution order is restricted by the constraint. |
| successor | EOCExecutableEntity RefAbstract | * | ref | The logical successor of an executable entity or a group of executable entities. |

Table 2.71: EOCExecutableEntityRef

| Class | EOCEventRef | | | |
|----------------------|--|-------|------|---|
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint::ExecutionOrderConstraint | | | |
| Note | This is used to define a reference to an RTE or BSW Event. | | | |
| Base | <i>ARObject</i> , EOCExecutableEntityRefAbstract , Identifiable , MultilanguageReferrable , Referrable | | | |
| Aggregated by | <i>ExecutionOrderConstraint.orderedElement</i> | | | |
| Attribute | Type | Mult. | Kind | Note |
| bswModule Instance | BswImplementation | 0..1 | ref | Specifies the BSW module instance the BSW event is related to. |
| component | SwComponent Prototype | 0..1 | iref | <p>This association references the specific instance of the SW-C prototype.</p> <p>InstanceRef implemented by: ComponentInCompositionInstanceRef</p> |
| event | AbstractEvent | 0..1 | ref | The AbstractEvent (event) whose execution order is restricted by the constraint. |
| successor | EOCExecutableEntity RefAbstract | * | ref | The logical successor of an executable entity or a group of executable entities. |

Table 2.72: EOCEventRef

The elements described above can be used for creating different patterns of [ExecutionOrderConstraints](#) for various purposes. These patterns are described in the following subsections. The constraints listed below applied to all of these patterns.

[constr_4525] Precedence of successor relationships `successor` and `directSuccessor`

Imposition time: [IT_SwcTd](#)

[The successor relationships `successor` and `directSuccessor` take always precedence over the ordered multiplicity of the association `nestedElement`.]

[constr_4532] Successor relationship is not self-referencing

Imposition time: [IT_SwcTd](#)

[The target and source of the successor relationships `successor` and `directSuccessor` shall not be the same. In other words an `EOCExecutableEntityRef` and `EOCExecutableEntityRefGroup` shall not reference itself as its logical or direct successor.]

[constr_4533] Maximum number of successor relationships*Imposition time:* **IT_SwcTd**

[The maximum number of successor relationships, namely **successor** or **direct-Successor**:

- between two **EOCExecutableEntityRefs**
- between two **EOCEventRefs**
- between two **EOCExecutableEntityRefGroups**
- between an **EOCExecutableEntityRef** and an **EOCExecutableEntityRefGroup**
- between an **EOCEventRef** and an **EOCExecutableEntityRefGroup**

is one (1).]

[constr_4534] Maximum number of **directSuccessor relationships***Imposition time:* **IT_SwcTd**

[The number of **directSuccessor** relationships of a:

- **EOCExecutableEntityRef**
- **EOCEventRef**
- **EOCExecutableEntityRefGroup**

shall not exceed the number of independent execution units available in a system.]

[constr_4536] Compatible recurrence of any **ExecutableEntity***Imposition time:* **IT_SwcTd**

[In an **ExecutionOrderConstraint** the **ExecutableEntitys**, referenced by all **EOCExecutableEntityRefs** respectively all **EOCEventRefs**, shall be compatible with regard to their recurrence.]

[constr_4537] References among elements in an **ExecutionOrderConstraint***Imposition time:* **IT_SwcTd**

[An **EOCExecutableEntityRef** respectively **EOCEventRef** or an **EOCExecutableEntityRefGroup** shall reference only **EOCExecutableEntityRefs**, respectively all **EOCEventRefs**, or **EOCExecutableEntityRefGroups** which are part of the same **ExecutionOrderConstraint**.]

[constr_4545] Referring either ExecutableEntitys or AbstractEvents*Imposition time:* IT_SwcTd

〔An `ExecutionOrderConstraint` shall contain either only `EOCExecutableEntityRef` or only `EOCEventRef`, but not both. In the former case `ExecutableEntitys` are referenced and in the latter case `AbstractEvents` are referenced.〕

[constr_4546] Setting the attribute isEvent*Imposition time:* IT_SwcTd

〔The value of the attribute `isEvent` shall be set to:

- TRUE: when the `ExecutionOrderConstraint` refers only to `AbstractEvents`
- FALSE: when the `ExecutionOrderConstraint` refers only to `ExecutableEntitys`

as per [constr_4545]

[constr_4547] Restriction of `ExecutionOrderConstraint.permitMultipleReferencesToEE`*Imposition time:* IT_SwcTd

〔The attribute `permitMultipleReferencesToEE` shall exist only if `ExecutionOrderConstraint.isEvent==FALSE` as per [constr_4546]〕

Rationale for [constr_4547]: The same `ExecutableEntity` may only be referenced more than once in case the `ExecutionOrderConstraint` shall solely refer to `ExecutableEntitys` (and not also `AbstractEvents`).

[constr_6833] Existence of `ExecutionOrderConstraint.orderedElement`*Imposition time:* IT_SubClasTdEvAss

〔For each `ExecutionOrderConstraint`, the attribute `orderedElement` shall exist at least once〕

[constr_6834] Existence of `EOCExecutableEntityRefGroup.nestedElement`*Imposition time:* IT_SubClasTdEvAss

〔For each `EOCExecutableEntityRefGroup`, the reference in the role `nestedElement` shall exist at least once〕

2.4.3.1.1 Ordinary Execution Order Constraint

A `ExecutionOrderConstraint.executionOrderConstraintType==ordinaryEOC` is used to specify an order of execution of `ExecutableEntity`s.

As shown in Figure 2.58 the `ExecutionOrderConstraint` contains a number of `EOCExecutableEntityRefs` which reference the `ExecutableEntity`s the execution order is imposed on. The associations `successor` and `directSuccessor` are used to specify the type of successor relationship and enforce the order of execution.

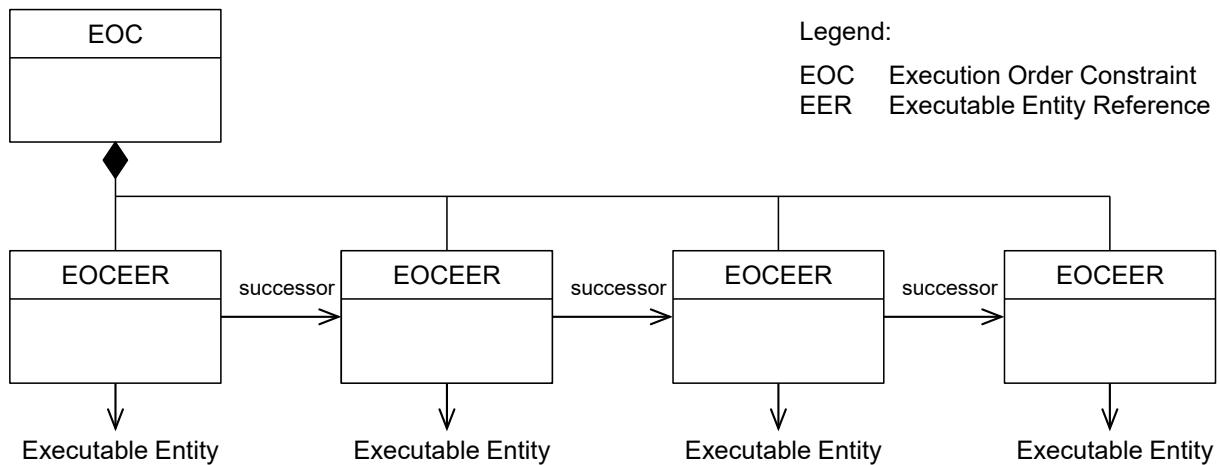


Figure 2.58: Example of an `ExecutionOrderConstraint` of type `ordinaryEOC`

In Figure 2.59, the `ExecutionOrderConstraint` contains a number of `event` references which refer to the specific `RTEEvents/BswEvents` the execution order is imposed on.

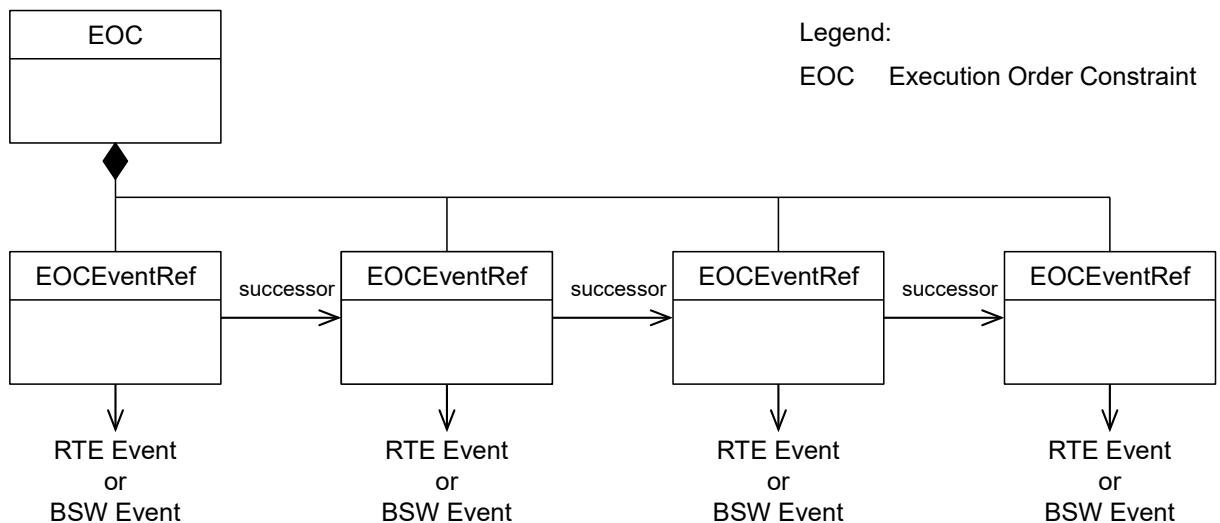


Figure 2.59: Example of an `ExecutionOrderConstraint` of type `ordinaryEOC` referencing `RTEEvents/BswEvents`

[constr_4541] Existence of `EOCExecutableEntityRef.executable` in an Ordinary Execution Order Constraint

Imposition time: `IT_SwcTd`

〔In an `ExecutionOrderConstraint.executionOrderConstraintType==ordinaryEOC`, in the tree of `orderedElements`, if the `orderedElement` is a `EOCExecutableEntityRef`, it shall reference an `ExecutableEntity` in the role `executable`.〕

[constr_4548] Existence of `EOCEventRef.event` in an Ordinary Execution Order Constraint

Imposition time: `IT_SwcTd`

〔In an `ExecutionOrderConstraint.executionOrderConstraintType==ordinaryEOC`, in the tree of `orderedElements`, if the `orderedElement` is a `EOCEventRef`, it shall reference an `AbstractEvent` in the role `event`.〕

2.4.3.1.2 Hierarchical Execution Order Constraint

A `ExecutionOrderConstraint.executionOrderConstraintType==hierarchicalEOC` specifies an order of execution of `ExecutableEntity`s using the capability of creating groups of `ExecutableEntity`s. In other words, it enables to specify tree-like structures of `EOCExecutableEntityRefGroup`s, `EOCExecutableEntityRef`s and `EOCEventRef`s.

As shown in Figure 2.60 the `ExecutionOrderConstraint` contains a number of `EOCExecutableEntityRef`s and one `EOCExecutableEntityRefGroup`, which in turn reference a number of `EOCExecutableEntityRef`s. The associations `successor` and `directSuccessor` between these `EOCExecutableEntityRef`s and the `EOCExecutableEntityRefGroup` are used to specify the type of successor relationship and enforce the order of execution.

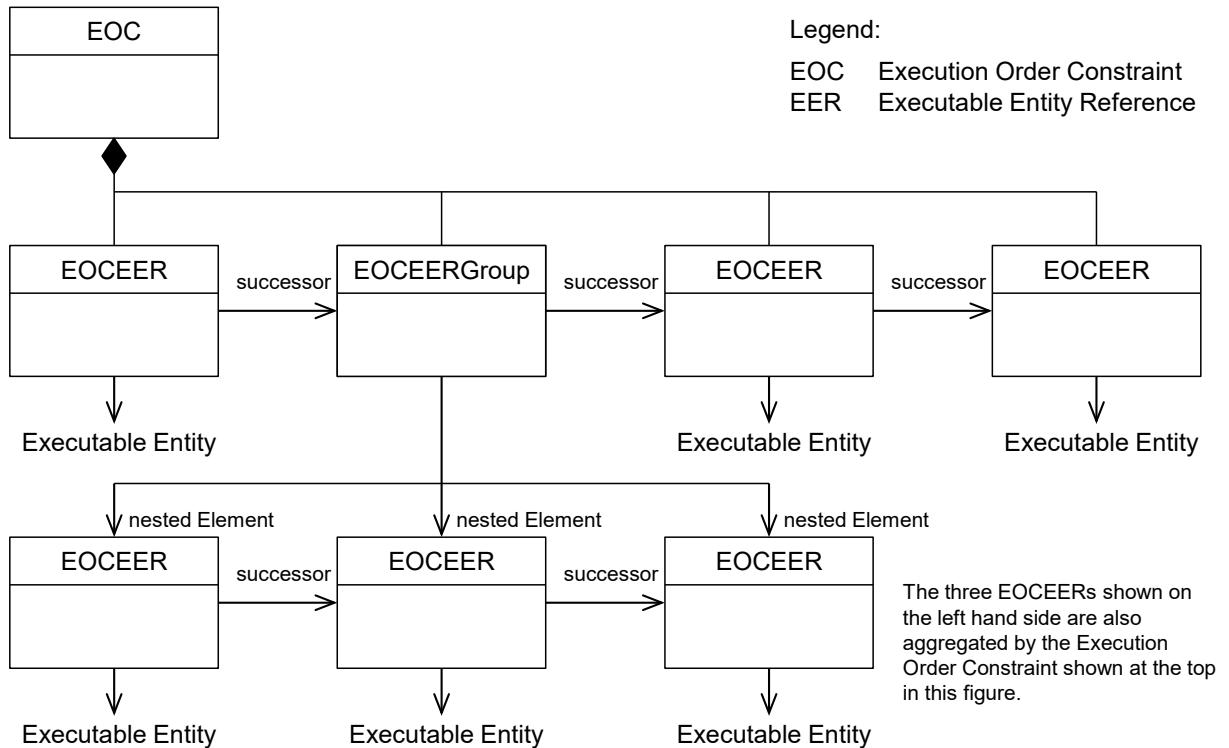


Figure 2.60: Example of a [ExecutionOrderConstraint](#) of type [hierarchicalEOC](#)

[constr_4523] Restriction of [maxCycleRepetitions](#) and [maxSlotsPerCycle](#) to Repetitive Execution Order Constraint

Imposition time: [IT_SwcTd](#)

〔The attributes

- [EOCExecutableEntityRefGroup.maxCycleRepetitions](#)
- [EOCExecutableEntityRefGroup.maxSlotsPerCycle](#)

shall exist only if *that EOCExecutableEntityRefGroup is aggregated by an [ExecutionOrderConstraint.executionOrderConstraintType==repetitiveEOC](#) in the role [orderedElement](#)*〕

[constr_4538] Hierarchical Execution Order Constraint: [EOCExecutableEntityRef](#), [EOCEventRef](#), and [EOCExecutableEntityRefGroup](#) shall be target or source of a successor relationship

Imposition time: [IT_SwcTd](#)

〔In a [ExecutionOrderConstraint.executionOrderConstraintType==hierarchicalEOC](#), each:

- [EOCExecutableEntityRefGroup](#)
- [EOCExecutableEntityRef](#)
- [EOCEventRef](#)

which is not part of an `EOCExecutableEntityRefGroup` shall be target or source of at least one successor relationship.]

[constr_4542] Existence of `EOCExecutableEntityRef.executable` in a Hierarchical Execution Order Constraint

Imposition time: `IT_SwcTd`

[In an `ExecutionOrderConstraint.executionOrderConstraintType==hierarchicalEOC`, in the tree of `orderedElements`, if the `orderedElement` is a `EOCExecutableEntityRef`, it shall reference an `ExecutableEntity` in the role `executable`.]

[constr_4549] Existence of `EOCEventRef.event` in a Hierarchical Execution Order Constraint

Imposition time: `IT_SwcTd`

[In an `ExecutionOrderConstraint.executionOrderConstraintType==hierarchicalEOC`, in the tree of `orderedElements`, if the `orderedElement` is a `EOCEventRef`, it shall reference an `AbstractEvent` in the role `event`.]

[TPS_TIMEX_00129] Representation of the root in a Hierarchical Execution Order Constraint

Status: DRAFT

[In a `ExecutionOrderConstraint.executionOrderConstraintType==hierarchicalEOC` the root in the hierarchy shall be the `EOCExecutableEntityRefGroup.category==ROOT_GROUP`.]

[constr_6909] Singleton `ROOT_GROUP` in a Hierarchical Execution Order Constraint

Status: DRAFT

Imposition time: `IT_SwcTd`

[In a `ExecutionOrderConstraint.executionOrderConstraintType==hierarchicalEOC`, in the tree of `orderedElements`, there shall be only **one** `EOCExecutableEntityRefGroup.category==ROOT_GROUP`]

[constr_6910] Referencing from a `ROOT_GROUP` in a Hierarchical Execution Order Constraint

Status: DRAFT

Imposition time: `IT_SwcTd`

[In a `ExecutionOrderConstraint.executionOrderConstraintType==hierarchicalEOC`, in the tree of `orderedElements`, in the singularly identifiable `ROOT_GROUP` (according to [constr_6909]):

- the `successor` shall not exist

- the `directSuccessor` shall not exist

]

[constr_6911] Referencing to a `ROOT_GROUP` in a Hierarchical Execution Order Constraint*Status:* DRAFT*Imposition time:* IT_SwcTd

[In a `ExecutionOrderConstraint.executionOrderConstraintType==hierarchicalEOC`, in the tree of `orderedElements`, the singularly identifiable `ROOT_GROUP` (according to [constr_6909]) shall not be referenced in the role:

- `successor` by another `EOCExecutableEntityRefGroup`
- `directSuccessor` by another `EOCExecutableEntityRefGroup`
- `nestedElement` by another `EOCExecutableEntityRefGroup`

]

Note: [constr_4537] forbids that the same subclass of `EOCExecutableEntityRefAbstract` is used in multiple `ExecutionOrderConstraints`.

2.4.3.1.3 Repetitive Execution Order Constraint

The `ExecutionOrderConstraint.executionOrderConstraintType==repetitiveEOC` is used to specify *varying* `ExecutionOrderConstraints` depending on subsequent occurrences of a specific event. This enables one to specify that specific `ExecutionOrderConstraints` are imposed on a given number of `ExecutableEntity`s whenever the particular `TimingDescriptionEvent` occurs.

For example:

- if the event A occurs:
 - the 1st time; the `ExecutableEntity`s: one (1), two (2) and three (3)
 - the 2nd time; the `ExecutableEntity`s: one (1), four (4) and five (5)
 - the 3rd time; the `ExecutableEntity`s: one (1), two (2) and three (3)
 - the 4th time; the `ExecutableEntity`s: one (1), four (4) and five (5)
 - ...
- ...shall be executed in this given order

The occurrences of the specified event are called *cycles* and the order of the `ExecutableEntity`s within a cycle is arranged by *slots*.

As shown in Figure 2.61 the Repetitive Execution Order Constraints follows a specific pattern.

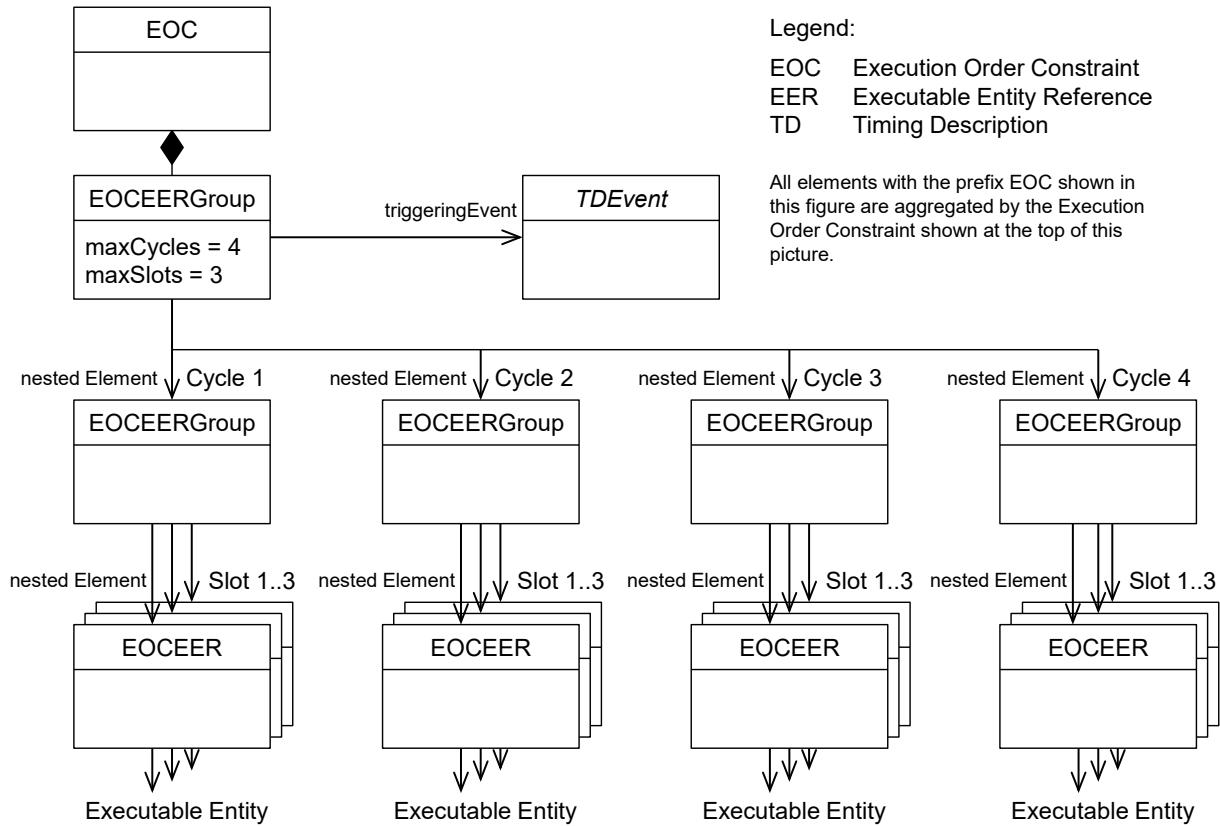


Figure 2.61: Example of a Repetitive Execution Order Constraint

The `executionOrderConstraintType==repetitiveEOC` follows the pattern of the Hierarchical Execution Order Constraint, but some restrictions apply to the use and structure of groups of executable entity references. The `ExecutionOrderConstraint` consists of one group of executable entity references, called the *root* group, which references only other groups of executable entity references.

These groups in turn reference executable entity references respectively event references which eventually reference the specific `ExecutableEntity`s respectively events. The *root* group specifies the maximum number of cycles `maxCycles` and the maximum number of slots `maxSlots`.

The `maxCycles` specifies the number of subsequent event occurrences after which the execution order constraint repeats, hence the name Repetitive Execution Order Constraint; the `maxSlots` specifies the number of `ExecutableEntity`s that are executed in a given order within a cycle.

Note: `maxCycles`, respectively *cycle*, is a synonym for the term *maxRepetitions*, respectively *repetition*.

The table below presents the repetitive execution order constraint shown in Figure 2.61 in a tabular way.

| Cycle | Slot 1 | Slot 2 | Slot 3 |
|--------|--------|--------|--------|
| Cycle1 | RE1 | RE2 | RE3 |
| Cycle2 | RE1 | RE4 | RE5 |
| Cycle3 | RE7 | RE8 | RE9 |
| Cycle4 | RE1 | RE8 | RE6 |

Table 2.73: Example Repetitive Execution Order Constraint

The following constraints shall be considered when creating Repetitive Execution Order Constraints:

[constr_4526] Specifying `maxCycles` and `maxSlots` in a Repetitive Execution Order Constraint

Imposition time: `IT_SwcTd`

〔The attributes `maxCycles` and `maxSlots` shall be specified only by the *root* group of executable entity references `EOCExecutableEntityRefGroup`.〕

[constr_4540] `maxCycles` and `maxSlots` shall not be zero

Imposition time: `IT_SwcTd`

〔If the attributes `maxCycles` and `maxSlots` are used, then the values of the attributes `maxCycles` and `maxSlots` shall be greater than zero (0).〕

[constr_4527] Referencing `TimingDescriptionEvent` in a Repetitive Execution Order Constraint

Imposition time: `IT_SwcTd`

〔The `TimingDescriptionEvent` shall be specified only by the *root* group of executable entity references `EOCExecutableEntityRefGroup`.〕

[constr_4528] The *root* `EOCExecutableEntityRefGroup` shall reference only `EOCExecutableEntityRefGroups`

Imposition time: `IT_SwcTd`

〔The *root* `EOCExecutableEntityRefGroup` shall reference only groups of executable entity references respectively event references grouped by the element `EOCExecutableEntityRefGroups`.〕

[constr_4529] Number of nested elements referenced by the *root* `EOCExecutableEntityRefGroup`

Imposition time: `IT_SwcTd`

〔The number of nested elements referenced by the *root* `EOCExecutableEntityRefGroup` shall be exactly the number given by the attribute `maxCycles`.〕

[constr_4530] An `EOCExecutableEntityRefGroup` representing a cycle shall reference only `EOCExecutableEntityRef`s respectively `EOCEventRef`s

Imposition time: `IT_SwcTd`

〔The `EOCExecutableEntityRefGroup` representing a cycle shall reference only `EOCExecutableEntityRef`s, respectively `EOCEventRef`s.〕

[constr_4531] Number of nested elements referenced by `EOCExecutableEntityRefGroup` representing a cycle

Imposition time: `IT_SwcTd`

〔The number of nested elements referenced by a `EOCExecutableEntityRefGroup` representing a cycle shall be exactly the number given by the attribute `maxSlots`.〕

[constr_4539] The successor relationships `successor` and `directSuccessor` shall not be used

Imposition time: `IT_SwcTd`

〔The successor relationships `successor` and `directSuccessor` shall not be used in a `executionOrderConstraintType==repetitiveEOC`.〕

[constr_6907] Restriction of `EOCExecutableEntityRefGroup.triggeringEvent`

Status: DRAFT

Imposition time: `IT_SwcTd`

〔The `TimingDescriptionEvent` referenced in the role `EOCExecutableEntityRefGroup.triggeringEvent` shall exist only if the `EOCExecutableEntityRefGroup` is transitively aggregated by an `ExecutionOrderConstraint.executionOrderConstraintType==repetitiveEOC` in the role `orderedElement`〕

2.4.3.2 ExecutionTimeConstraint

An AUTOSAR `ResourceConsumption.executionTime` can be used to describe the execution time of an `ExecutableEntity`. The concept is described in [9]. This execution time description represents a timing property of a `ExecutableEntity`.

| | | | | |
|----------------------|--|--------------|-------------|---|
| Class | ExecutionTimeConstraint | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint::ExecutionTimeConstraint | | | |
| Note | <p>Constrains the execution time of the referenced <code>executable</code> in <code>component</code> between a <code>minimum</code> and <code>maximum</code> interval.</p> <p>The time to execute the <code>executable</code> including interruptions by other entities and including external calls is commonly called "response time". The TimingExtensions provide the concept of event chains and latency constraints for that purpose. An event chain from the start of the entity to the termination of the entity with according latency constraint represents a response time constraint for that executable entity.</p> | | | |
| Base | <i>ARObject, Identifiable, MultilanguageReferrable, Referrable, TimingConstraint, Traceable</i> | | | |
| Aggregated by | <i>TimingExtension.timingGuarantee, TimingExtension.timingRequirement</i> | | | |
| Attribute | Type | Mult. | Kind | Note |
| component | <code>SwComponentPrototype</code> | 0..1 | iref | <p>The component that contains the referenced Executable Entity for the ExecutionTimeConstraint. If the entity is in a basic software module no component shall be provided.</p> <p>InstanceRef implemented by: <code>ComponentInCompositionInstanceRef</code></p> |
| executable | <code>ExecutableEntity</code> | 0..1 | ref | The referenced ExecutableEntity for the ExecutionTime Constraint. |
| executionTimeType | <code>ExecutionTimeTypeEnum</code> | 0..1 | attr | Specifies the type of the execution time constrained by ExecutionTimeConstraint, |
| maximum | <code>MultidimensionalTime</code> | 0..1 | aggr | The maximum execution time. |
| minimum | <code>MultidimensionalTime</code> | 0..1 | aggr | The minimum execution time. |

Table 2.74: ExecutionTimeConstraint

| | |
|----------------------|---|
| Enumeration | ExecutionTimeTypeEnum |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint::ExecutionTimeConstraint |
| Note | Specifies the type of the <code>executionTimeType</code> for a <code>ExecutionTimeConstraint</code> . |
| Aggregated by | <code>ExecutionTimeConstraint.executionTimeType</code> |
| Literal | Description |
| gross | Indicates that the given execution time is the time used to execute the <code>executable</code> WITHOUT any interruption and WITH external calls. Tags: <code>atp.EnumerationLiteralIndex=0</code> |
| net | Indicates that the given execution time is the time used to execute the <code>executable</code> WITHOUT any interruption and WITHOUT any external calls. Tags: <code>atp.EnumerationLiteralIndex=1</code> |

Table 2.75: ExecutionTimeTypeEnum

[TPS_TIMEX_00008] `ExecutionTimeConstraint` to specify execution time constraints

Upstream requirements: RS_TIMEX_00001, RS_TIMEX_00013

〔The element `ExecutionTimeConstraint` is used to specify minimum and maximum execution time constraints of `ExecutableEntity`s〕

An `ExecutionTimeConstraint` references the `ExecutableEntity`, whose execution time shall be constrained. The `ComponentInCompositionInstanceRef` referenced by `component` defines the component instance, which contains the `RunnableEntity` (in case of a BSW `ExecutableEntity`, the `component` reference is omitted).

[constr_6835] Existence of `ExecutionTimeConstraint.executionTimeType`

Imposition time: `IT_SwcTd`

「For each `ExecutionTimeConstraint`, the attribute `executionTimeType` shall exist」

[constr_6836] Existence of `ExecutionTimeConstraint.executable`

Imposition time: `IT_SwcTd`

「For each `ExecutionTimeConstraint`, the reference to `ExecutableEntity` in the role `executable` shall exist」

2.4.3.3 SynchronizationPointConstraint

The capabilities of an `ExecutionOrderConstraint` are not sufficient to *explicitly* express that the execution of one or more `ExecutableEntity`s shall be started if and only if one or more other `ExecutableEntity`s have finished execution. In other words, one or more `ExecutableEntity`s shall succeed the execution of one or more other `ExecutableEntity`s when those `ExecutableEntity`s have all finished their execution. In order to specify this behavior the `SynchronizationPointConstraint` is used which specifies synchronization points between `ExecutableEntity`s, referenced by their corresponding `RTEEvents` or `BswEvents`, or `EOCEExecutableEntityRefGroup`s.

Note that the element `SynchronizationTimingConstraint` is also related to synchronization, but this element imposes *temporal* constraints on the occurrence of events, like the occurrence of activation, start and termination of an executable entity. For example, a number of `ExecutableEntity`s shall terminate and a number of executable entities shall start within a given time interval. But it is not possible to specify that the `ExecutableEntity`s shall terminate in this time interval *before* the `ExecutableEntity`s start in this time interval. Therefore, this capability is not sufficient to express a synchronization point.

Indeed, it is possible to combine the two constraints, `SynchronizationPointConstraint` and `SynchronizationTimingConstraint` in order to specify that a synchronization point between a number of `ExecutableEntity`s is mandatory and these `ExecutableEntity`s shall terminate and start within a given time interval.

| | |
|----------------------|--|
| Class | <code>SynchronizationPointConstraint</code> |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint::SynchronizationPointConstraint |
| Note | Specifies a synchronization point either between groups of <code>ExecutableEntity</code> s or individual <code>ExecutableEntity</code> s referenced via their corresponding RTE or BSW events. |
| Base | <code>ARObject</code> , <code>Identifiable</code> , <code>MultilanguageReferrable</code> , <code>Referrable</code> , <code>TimingConstraint</code> , <code>Traceable</code> |
| Aggregated by | <code>TimingExtension.timingGuarantee</code> , <code>TimingExtension.timingRequirement</code> |





| Class | SynchronizationPointConstraint | | | |
|-------------|---------------------------------|-------|------|---|
| Attribute | Type | Mult. | Kind | Note |
| sourceEec | EOCExecutableEntity RefGroup | * | ref | The source executable entities cluster containing the executable entities that shall finish execution before the synchronization point. |
| sourceEvent | AbstractEvent | * | ref | The executable entities --- referenced by their events --- that shall finish execution before the synchronization point. |
| targetEec | EOCExecutableEntity RefGroup | * | ref | The target executable entities cluster containing the executable entities that shall start execution after the synchronization point. |
| targetEvent | AbstractEvent | * | ref | The executable entities --- referenced by their events --- that shall start execution after the synchronization point. |

Table 2.76: SynchronizationPointConstraint

[TPS_TIMEX_00054] *SynchronizationPointConstraint* explicitly specifies a synchronization point between executable entities

Upstream requirements: RS_TIMEX_00023

[The element *SynchronizationPointConstraint* is used to specify synchronization points between *ExecutableEntitys*.]

3 Application

This chapter presents how aspects in [2](#) are combined to achieve particular higher-level application of timing.

3.1 Logical Execution Time

[Logical Execution Time \(LET\)](#) is a real-time programming *abstraction*. It abstracts from actual (*physical*) execution time of a real-time program. LET determines the time it takes from reading program input to writing program output regardless of the time it takes to execute the program which processes the input and writes output. LET is motivated by the observation that the relevant behavior of real-time programs is determined by when input is read and output is written and not when programs just execute any code.

This *logical* notion is shown in the upper part of Figure [3.1](#). At the *release* point input is read and at the *terminate* point output is written. Between these two points — the logical execution time — the program is executed, and writing output is delayed until the logical execution time elapsed even if the program completes execution before the logical execution time elapsed. This is sketched in the lower part of Figure [3.1](#) and shows the actual *physical* execution of a program.

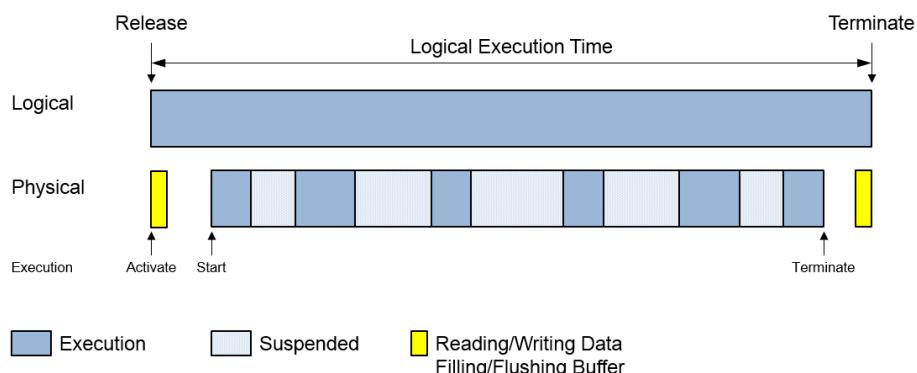


Figure 3.1: LET Interval

3.1.1 LET Interval Definition

Modeling of LET starts with a categorized [TimingDescriptionEventChain](#). The [TimingDescriptionEventChain.category==LET_INTERVAL](#) to represent the LET interval span. This [TimingDescriptionEventChain](#) shall reference two [TDEventComplex](#)s to represent the *stimulus* (see [[TPS_TIMEX_00111](#)]) and the *response* (see [[TPS_TIMEX_00114](#)]) points.

The `TDEventComplex` is used because both the `stimulus` and `response` are not referencing an *observable* location, like the other types of `TimingDescriptionEvent`s. A `TDEventOccurrenceExpression` on the `TDEventComplex` is used to specify the particular `TimingDescriptionEvent` supposed to occur in case of the release and terminate point respectively.

[TPS_TIMEX_00057] Representation of an LET interval in a `TimingDescriptionEventChain`

Upstream requirements: [RS_TIMEX_00022](#)

〔An LET Interval shall be described by a `TimingDescriptionEventChain.category==LET_INTERVAL`〕

[TPS_TIMEX_00112] Representation of an LET interval `release`

Status: DRAFT

Upstream requirements: [RS_TIMEX_00022](#)

〔An LET Interval release shall be described by `stimulus` referencing a `TDEventComplex.category==LET_RELEASE`〕

[TPS_TIMEX_00113] Representation of an LET interval `terminate`

Status: DRAFT

Upstream requirements: [RS_TIMEX_00022](#)

〔An LET Interval terminate shall be described by `response` referencing a `TDEventComplex.category==LET_TERMINATE`〕

After defining the `stimulus` and `response` events, the timing related parameters of the an LET interval are specified using `TimingConstraints`. A `LatencyTimingConstraint` specifies the **duration** and a `PeriodicEventTriggering` specifies the **recurrence** of an LET interval.

[TPS_TIMEX_00121] Representation of the duration of an LET interval

Status: DRAFT

〔The **duration** of an LET interval shall be described by a `LatencyTimingConstraint` where:

- `latencyConstraintType==reaction`
- `maximum` shall specify the **duration** semantics.

〕

[TPS_TIMEX_00117] Representation of the recurrence of an LET interval

Status: DRAFT

〔The **recurrence** of an LET interval shall be described by a [PeriodicEventTriggering](#) referencing a [TimingDescriptionEvent](#) in the role [event](#).〕

For LET, when expressing the duration semantics, only the [maximum](#) is needed, therefore the [minimum](#) shall be omitted from modeling.

[constr_6920] Existence of [LatencyTimingConstraint.minimum](#) used in an LET interval

Status: DRAFT

Imposition time: [IT_SwcTd](#)

〔For a [LatencyTimingConstraint](#) with:

- [latencyConstraintType==reaction](#)
- [scope.category==LET_INTERVAL](#)

the attribute [minimum](#) shall not exist.〕

It is only necessary to have a This [PeriodicEventTriggering](#) constraint shall be imposed on the [stimulus](#) event (the release point of the LET interval) of the timing description event chain representing a LET interval.

[TPS_TIMEX_00122] Application of a [PeriodicEventTriggering](#) constraint used to specify the **recurrence of an LET interval**

Status: DRAFT

〔For a [PeriodicEventTriggering](#) constraint used to configure the **recurrence** semantics of an LET interval, the reference [event](#) shall refer to the [TDEventComplex](#) used in the role [TimingDescriptionEventChain.stimulus](#).〕

[constr_6912] Mandatory specification of LET interval recurrence

Status: DRAFT

Imposition time: [IT_SwcTd](#)

〔For a [TDEventComplex.category==LET_RELEASE](#), there shall exist a [PeriodicEventTriggering](#) referencing that [TDEventComplex.category==LET_RELEASE](#) in the role [event](#)〕

Figure 3.2 shows an example LET model with ARXML listings [C.2](#), [C.3](#), [C.4](#).

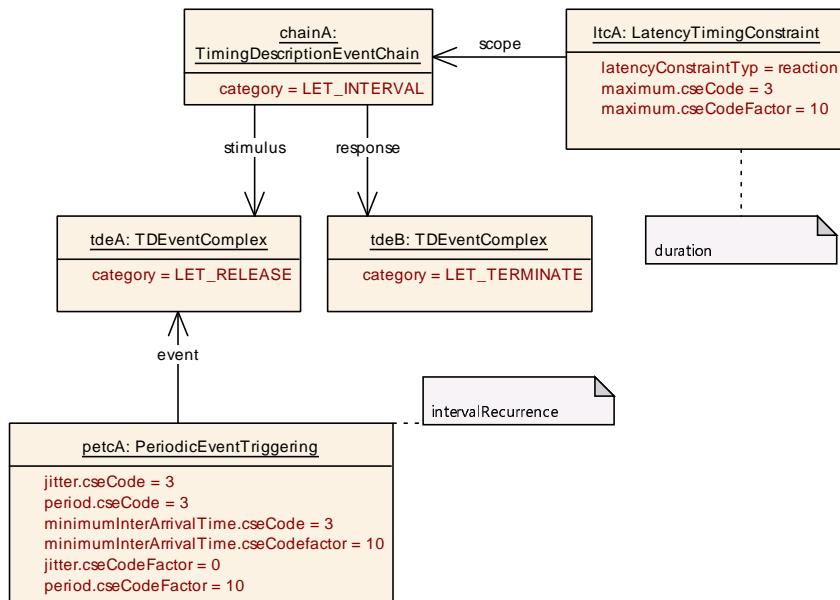


Figure 3.2: LET Interval Model

3.1.2 Relation between LET Intervals

This section describes the most prominent cases for describing LET intervals and possible relationships among them. These cases are frequently occurring in developing systems. The section describes various ways to use the AUTOSAR elements [PeriodicEventTriggering](#) and [OffsetTimingConstraint](#) to specify the duration of LET intervals and offsets between LET intervals. Offsets between LET intervals result in overlapping LET intervals or LET intervals in subsequent order with and without gaps between each other.

In Figure 3.3 a periodic LET interval is specified. The LET interval is periodically repeated and the value of the LET interval is the same as the period.

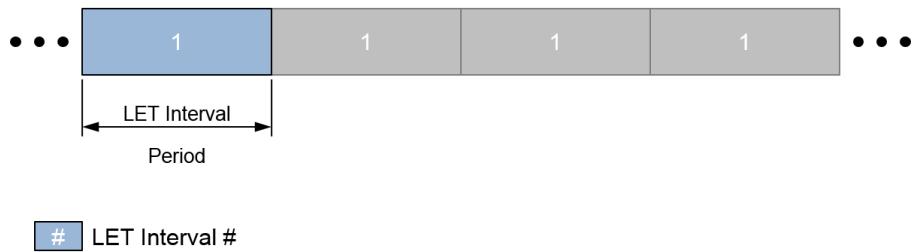


Figure 3.3: Specifying Periodic LET Interval

The corresponding ARXML fragments that describe this pattern are shown in listing C.2, C.3 and C.4. The first listing presents an ARXML fragment that specifies the LET interval; the second listing presents an ARXML fragment that specifies the duration of

the LET interval; and the third listing presents an ARXML fragment that specifies the recurrence of the LET interval.

In Figure 3.4 a periodic LET interval is specified, but the value of the period is larger than the LET interval. This results in a gap between the *terminate* point of the LET interval and the next *release* point of the LET interval. The corresponding ARXML fragment is shown in listing C.5. The first part of the ARXML fragments specifies the duration of the LET interval which is 6ms by using the latency timing constraint [LatencyTimingConstraint](#). The second part specifies the recurrence of the LET interval which is 10ms by using the periodic event triggering constraint [PeriodicEventTriggering](#).

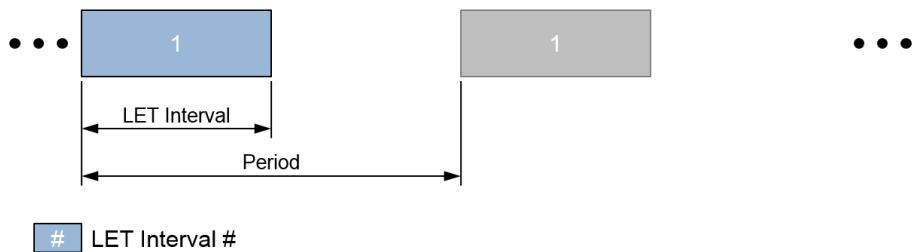


Figure 3.4: Specifying Periodic LET Interval with Gap

In Figure 3.5 three LET intervals are specified. The duration of the first LET interval 1 is 2.5ms, the duration of the second LET interval 2 is 5ms, and the duration of the third LET interval 3 is 10ms. The very first release of each LET interval takes place at the same point in time and then each of the LET intervals is repeated periodically. This case is modeled using the Offset Timing Constraint [OffsetTimingConstraint](#) and the offset between the *release* points of those LET intervals is 0ms, namely no offset.

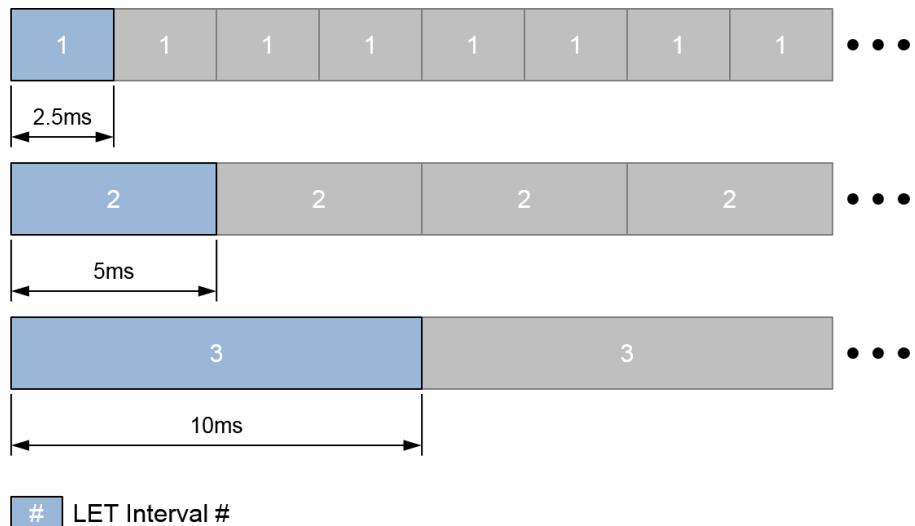


Figure 3.5: Specifying Relationship between Periodic LET Intervals

The listing C.6 shows an ARXML fragment which specifies two offset timing constraints between the *release* point of LET interval 1 and the two *release* points of the LET interval 2 and LET interval 3. In both cases the offset is zero (0ms) specifying that all LET intervals are released at the same point in time.

In Figure 3.6 three LET intervals are specified. The duration of the first LET interval 1 is 2.5ms, the duration of the second LET interval 2 is 5ms, and the duration of the third LET interval 3 is 10ms. The very first release of each LET interval takes place at different points in time and then each LET interval is repeated periodically. This case is modeled using the Offset Timing Constraint [OffsetTimingConstraint](#). The offset between the *release* points of the first LET interval 1 and second LET interval 2 is 2.5ms; and the offset between the *release* points of the first LET interval 1 and third LET interval 3 is 7.5ms.

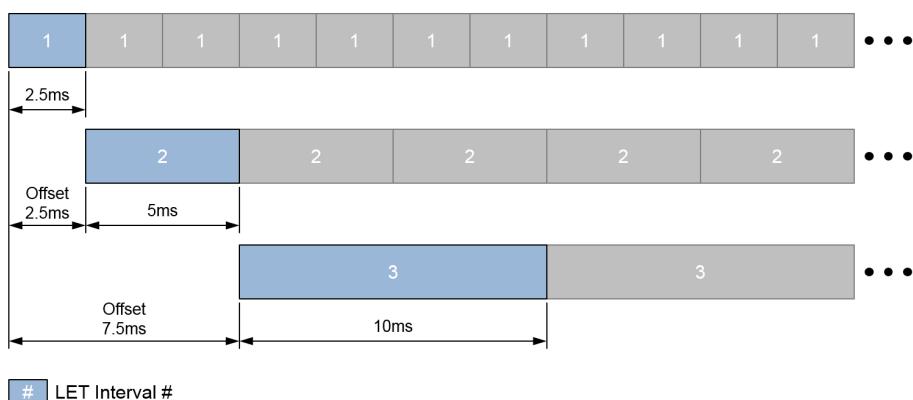


Figure 3.6: Specifying Relationship between Periodic LET Intervals With Offsets

The listing C.7 shows an ARXML fragment which specifies two offset timing constraints between the *release* point of LET interval 1 and the two *release* points of the LET interval 2 and LET interval 3. In the first case the offset between the *release* points of LET interval 1 and LET interval 2 is 2.5ms; and in the second case the offset between the *release* points of LET interval 1 and LET interval 3 is 7.5ms.

In Figure 3.7 four LET intervals are specified. The first LET interval 1 terminates before the second LET interval 2 is released; the second LET interval 2 terminates before the third LET interval 3 is released; and the third LET interval 3 terminates before the fourth LET interval 4 is released. After the last, the fourth LET interval 4 terminates, the same sequence of LET intervals is repeated. This case is modeled using the Offset Timing Constraint [OffsetTimingConstraint](#) to specify the offsets between the *release* point of the first LET interval 1 and the *release* points of each LET interval 2 through 4. The duration of each LET interval is listed in Table 3.1 along with the corresponding offsets between the *release* points of the LET intervals.

The periodic event triggering constraint [PeriodicEventTriggering](#) is used to specify that the period of the sequence is 32ms.

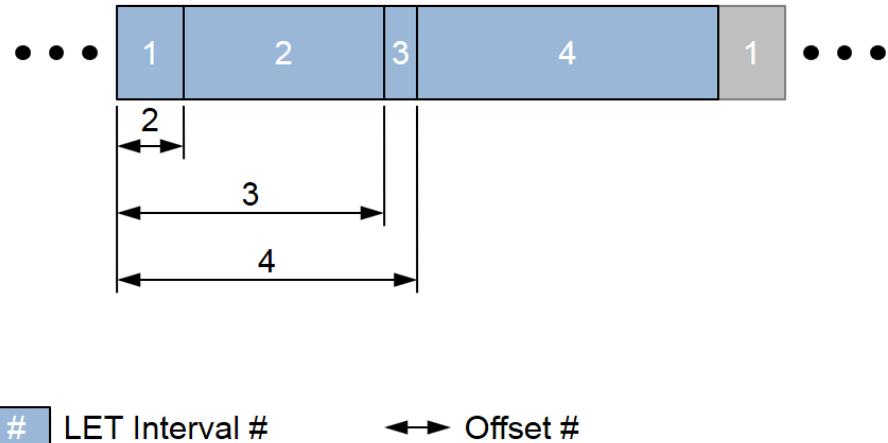
**Figure 3.7: Specifying Arbitrary LET Intervals**

Table 3.1 below summarizes the duration of the LET intervals and the offsets between the release point of the LET interval 1 and the release points of the LET intervals 2 through 4 as shown in Figure 3.7.

| LET Interval | Length | Offset |
|--------------|--------|-----------|
| 1 | 2ms | No offset |
| 2 | 6ms | 2ms |
| 3 | 1ms | 8ms |
| 4 | 9ms | 9ms |

Table 3.1: Specifying Arbitrary LET Intervals

The listing C.8 shows an ARXML fragment which specifies the three offsets between the *release* point of the first LET interval 1 and the *release* points of the three LET intervals 2 through 4.

In Figure 3.8 a similar case as described in Figure 3.7 is shown but the offset between some LET intervals and the duration of some LET intervals are different which result in some gaps between LET intervals. The duration of each LET interval and the offsets between the LET intervals of this case are listed in Table 3.2 along with the corresponding offsets between the *release* points of the LET intervals.

Like in the previous example the periodic event triggering constraint [PeriodicEvent-Triggering](#) is used to specify that the period of the sequence is 32ms.

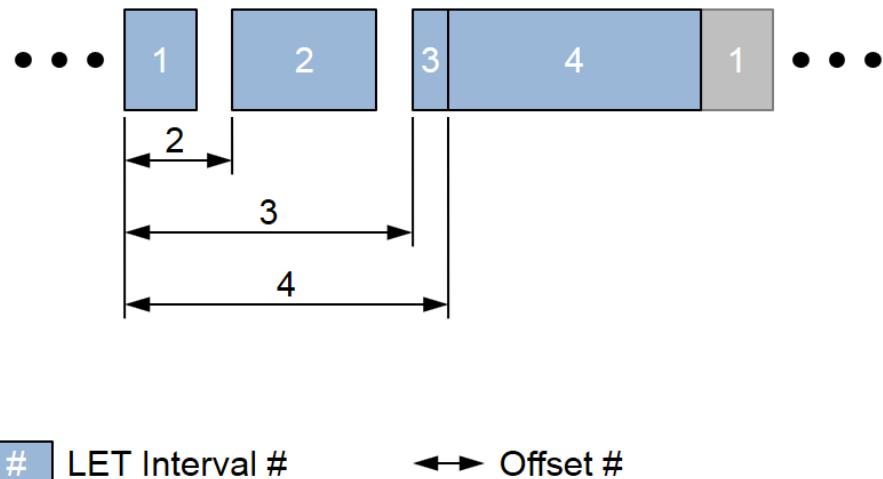


Figure 3.8: Specifying Arbitrary LET Intervals with Gaps

Table 3.2 below summarizes the duration of the LET intervals and the offsets between the release point of the LET interval 1 and the release points of the LET intervals 2 through 4 as shown in Figure 3.8.

| LET Interval | Length | Offset |
|--------------|--------|-----------|
| 1 | 2ms | No offset |
| 2 | 4ms | 3ms |
| 3 | 1ms | 8ms |
| 4 | 7ms | 9ms |

Table 3.2: Specifying Arbitrary LET Intervals with Gaps

The ARXML fragment of this example is the same as shown in listing C.8 except the values of the offset timing constraints, and the latency timing constraints, which specify the duration of the four LET intervals.

Considering the development of systems consisting of several electronic control units, each of them most likely equipped with multiple processors, as well as processors containing several independent execution units, also known as *cores*, there is the necessity to be able to specify LET intervals supporting the parallel execution of executable entities.

In Figure 3.9 four LET intervals are specified which overlap. The LET interval 3 and LET interval 4 are overlapping each other and overlapping LET interval 1 and LET interval 2, too.

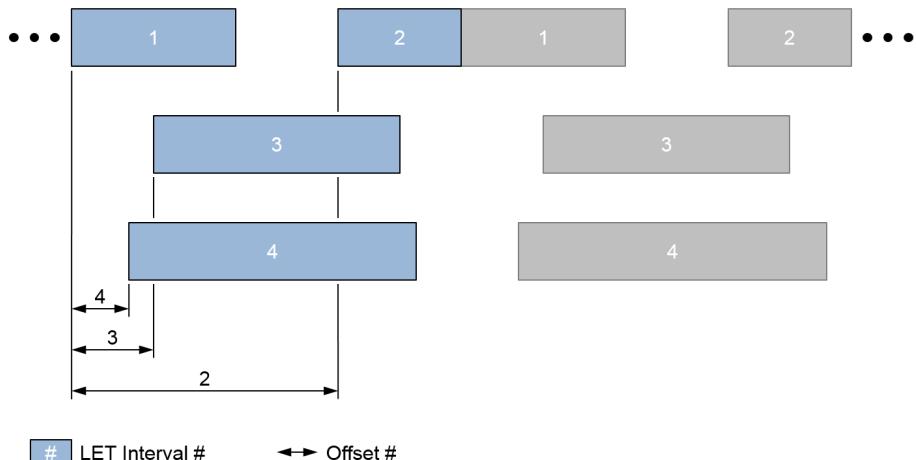
**Figure 3.9: Specifying Arbitrary LET Intervals with Overlap**

Table 3.3 below summarizes the duration of the LET intervals and the offsets between the release point of the LET interval 1 and the release points of the LET intervals 2 through 4 as shown in Figure 3.9.

| LET Interval | Length | Offset |
|--------------|--------|-----------|
| 1 | 8ms | No offset |
| 2 | 6ms | 13ms |
| 3 | 12ms | 4ms |
| 4 | 14ms | 3ms |

Table 3.3: Specifying Arbitrary LET Intervals with Overlap

The periodic event triggering constraint [PeriodicEventTriggering](#) is used to specify that the period of the sequence is 19ms.

The listing C.9 shows an ARXML fragment which specifies three offset timing constraints between the *release* point of LET interval 1 and the three *release* points of the LET interval 2 through LET interval 4. The values of the offsets are selected such that LET interval 3 and LET interval 4 are overlapping the LET interval 1 and LET interval 2. The offset between the *release* points of LET interval 1 and LET interval 2 is 13ms; the offset between the *release* point of LET interval 1 and LET interval 3 is 4ms; and the offset between the *release* point of LET interval 1 and LET interval 4 is 3ms.

3.1.3 Executable Entity Cluster to LET Interval mapping

During the development of distributed real-time systems there is frequently the case to be capable of grouping executable entities for various purposes, like scheduling,

mapping a number of executable entities to LET intervals, or specifying other constraint that are imposed on a group of executable entities¹.

The capability of the execution order constraint `ExecutionOrderConstraint` to group executable entities — references to the RTE or BSW events of those executable entities — is used to specify clusters of executable entities. In particular, an execution order constraint executable entity reference group `EOCEExecutableEntityRefGroup` (EOCEERG) is used to references all RTE and/or BSW events of the executable entities that are part of an *executable entities cluster*. And in turn an execution order constraint executable entity reference group can reference other groups of executable entities to form a hierarchy of executable entities clusters.

The term *Executable Entities Cluster* is a synonym for `EOCEExecutableEntityRefGroup` — also known as *group of executable entities*. An executable entities cluster is mapped to a LET interval using the attribute `letInterval`.

As an example, assume there are 8 x `AtomicSwComponentType`s: `Swc01 .. Swc08`. Each of these SWCs has an `internalBehavior` `ib_Swc0#` where the number sign indicates one of the SWCs `Swc01 .. Swc08`. Each SWC has 1 x `RunnableEntity` called `re_re01_Swc0#` and a Timing Event called `te_re01_Swc0#` is specified and references the corresponding `RunnableEntity` in the SWC.

The `RunnableEntity`s have data read and data write access points which read data from the ports of the particular SWC and write data to the provided ports of the particular SWC. The data dependency graph among the `RunnableEntity`s is shown in Figure 3.10. For this example three executable entities clusters are defined: The first executable entities cluster consists of the runnable entities `re_re01_Swc01`, `re_re01_Swc02` and `re_re01_Swc04`; the second one consists of the runnable entities `re_re01_Swc03` and `re_re01_Swc07`; and the third executable entities cluster consists of the runnable entities `re_re01_Swc05`, `re_re01_Swc06` and `re_re01_Swc08`. Three LET intervals are defined in the example and are called `LETinterval1`, `LETinterval2` and `LETinterval3`. The first executable entities cluster is mapped to the first LET interval using the association called `letInterval` between the element `EOCEExecutableEntityRefGroup` and the element `TimingDescriptionEventChain`.

¹The term "executable entity" is used to refer to both, AUTOSAR *Runnable Entity* and *Basic Software Module Entity*. Indeed, the AUTOSAR element `ExecutableEntity` is an abstract class and `RunnableEntity` and `BswModuleEntity` are specializations of this class.

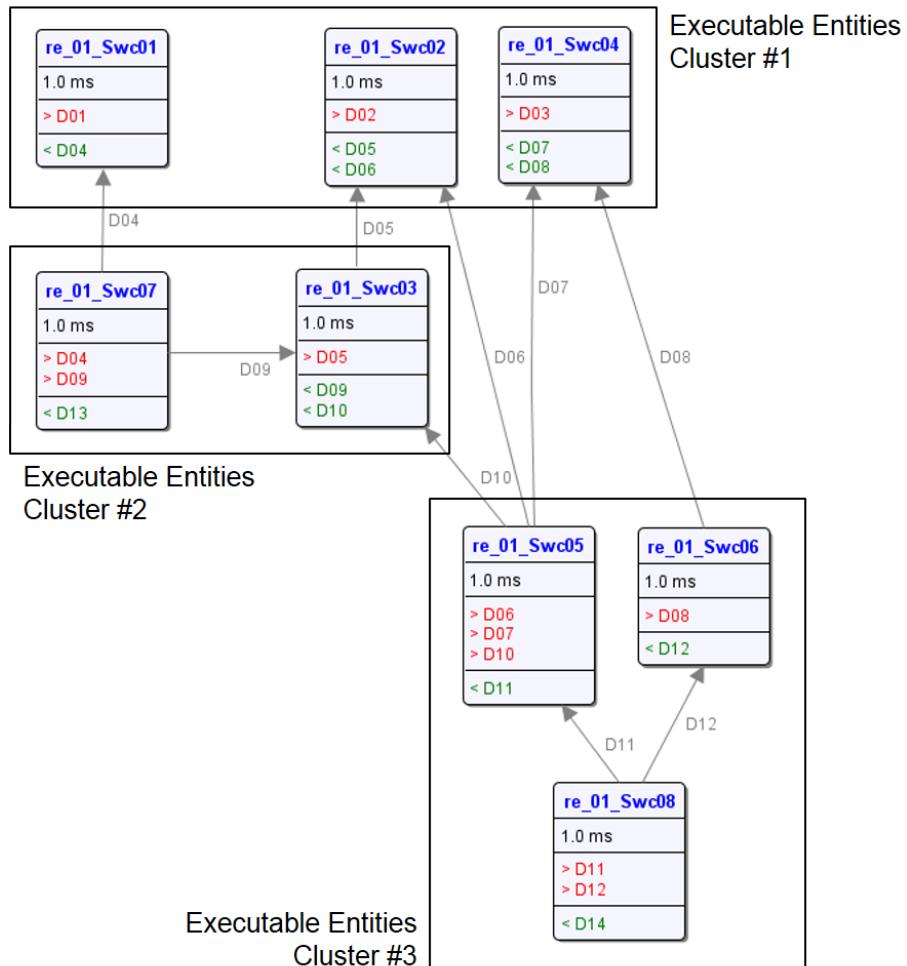


Figure 3.10: Data Dependencies among Executable Entities

The listing C.10 shows an ARXML fragment which specifies that an executable entities cluster and that it is mapped to the *LETinterval1*.

An executable entities cluster is mapped to a LET interval by letting the `EOCEExecutableEntityRefGroup` reference the timing description event chain that plays the role of a LET interval.

[constr_4554] Restriction of the referenced `TimingDescriptionEventChain` for a `letInterval`

Imposition time: `IT_SwcTd`

[The element `EOCEExecutableEntityRefGroup.letInterval` shall be present only in a `ROOT_GROUP` (according to [constr_6909])]

[constr_6913] Restriction on RTEEvents used in an LET interval

Status: DRAFT

Imposition time: IT_SwcTd

〔An `EOCExecutableEntityRefGroup` which references a `TimingDescription-EventChain.category==LET_INTERVAL` in the role `letInterval` **and** transitively references an `EOCEventRef` in the role `successor` **or** references an `EOCEventRef` in the role `nestedElement`, those `EOCEventRef`s shall reference either:

- `TimingEvent` in the role `event` **or**
- `BswTimingEvent` in the role `event`

〕

3.1.4 Data flow within an LET Interval

The `release` and `terminate` event of a `letInterval` define when input data is expected to be read and output data to be written. This provides a unique data flow for data exchange between `ExecutableEntity`s that are part of `EOCExecutableEntityRefGroups` which reference different LET intervals.

For all `ExecutableEntity`s in `EOCExecutableEntityRefGroups` that relate to a single `letInterval`, two paradigms are possible of how data is exchanged **within** an `letInterval` are possible - in Figure 3.11.

| | |
|---------------------------|---|
| Enumeration | <code>LetDataExchangeParadigmEnum</code> |
| Package | <code>M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint::ExecutionOrderConstraint</code> |
| Note | Specifies the data exchange paradigm between <code>ExecutableEntity</code> s within a LET interval. Tags: atp.Status=draft |
| Aggregated by | <code>EOCExecutableEntityRefGroup.letDataExchangeParadigm</code> |
| Literal | Description |
| <code>interLetOnly</code> | <p>All <code>ExecutableEntity</code>s mapped to this LET interval exchange data ONLY at the release and terminate event of the LET interval.</p> <p>This allows for a straightforward translation of the required label buffering but results in longer end-to-end latencies (multiple of the period). The execution order of <code>\ARMetaClass{Executable Entity}</code>s within the LET interval does not affect the data flow.</p> <p>Tags: <code>atp.EnumerationLiteralIndex=0</code> <code>atp.Status=draft</code></p> |



| Enumeration | LetDataExchangeParadigmEnum |
|-------------|---|
| intraLetEOC | <p>The ExecutableEntitys that belong to the same EOCExecutableEntityRefGroup and are mapped to this LET interval are executed in the order defined by the EOCExecutableEntityRefGroup and exchange data directly within this LET interval according to implicit semantics.</p> <p>Only at the borders of the LET interval or between independent EOCExecutableEntityRefGroups, is data propagated according to the LET paradigm.</p> <p>Tags: <code>atp.EnumerationLiteralIndex=1</code> <code>atp.Status=draft</code></p> |

Table 3.4: LetDataExchangeParadigmEnum

[TPS_TIMEX_00128] Default [letDataExchangeParadigm](#)

Status: DRAFT

〔The default [letDataExchangeParadigm](#)=[intraLetEOC](#).〕

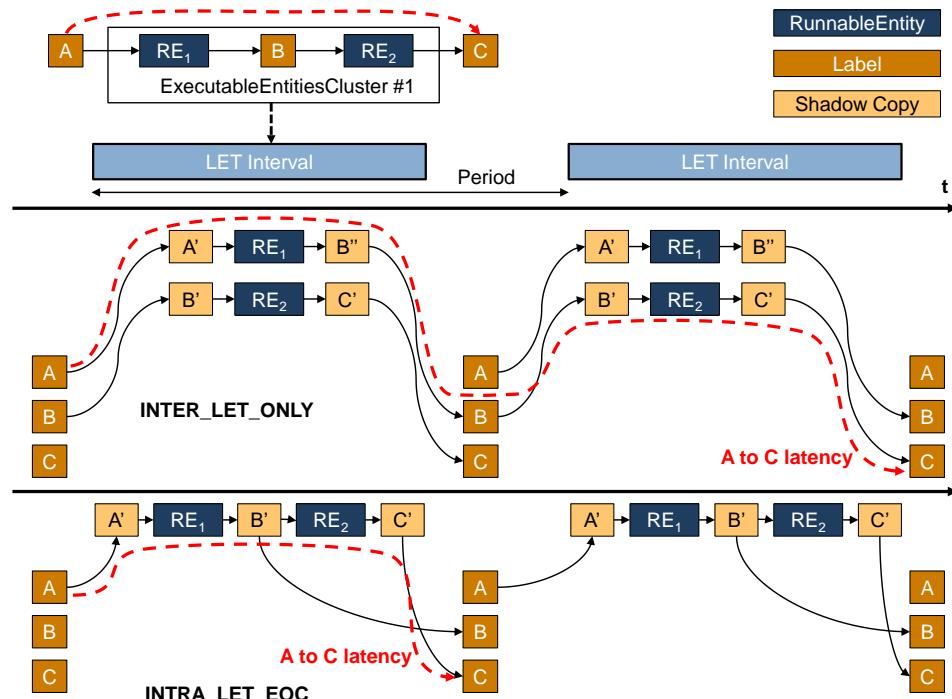


Figure 3.11: Data exchange paradigms within an LET interval

The [letDataExchangeParadigm](#) shall only apply to [EOCExecutableEntityRefGroups](#) used in the context of an LET interval and shall therefore be constrained to reflect that.

[constr_6908] Restriction of `EOCExecutableEntityRefGroup.letDataExchangeParadigm`

Status: DRAFT

Imposition time: IT_SwcTd

〔The attribute `letDataExchangeParadigm` shall exist only if the `letInterval` in the same same `EOCExecutableEntityRefGroup` references a `TimingDescriptionEventChain.category==LET_INTERVAL`〕

3.2 System Level Logical Execution Time

System Level Logical Execution Time (SL-LET) builds upon the concept of Logical Execution Time but targets a system-level view, e.g. between ports on (heterogenous) CP ECUs and/or AP Machines [10].

While the scope of [LET](#) in AUTOSAR is to specify a time interval for the execution of runnable entities (and therefore CP only) on a local ECU, SL-LET:

- addresses the data flow on arbitrary ports (e.g., ports of functional blocks, SWCs, services) and allows to specify latencies (resp. SL-LET interval durations) larger than the period,
- allows to incorporate distributed communication by explicitly addressing distributed clocks that have a bounded synchronization accuracy,
- can be applied already on the functional level and decomposed throughout the development process.

The assumption of instantaneous SL-LET events for read and write actions implies that those actions take place in logically zero time. A correct implementation of SL-LET must preserve the data flow semantics of the SL-LET model.

3.2.1 SL-LET Interval Definition

Modeling of SL-LET starts with a categorized [TimingDescriptionEventChain](#).

[TPS_TIMEX_00125] Representation of an SL-LET interval in a [TimingDescriptionEventChain](#)

Status: DRAFT

Upstream requirements: [RS_TIMEX_00026](#)

「An SL-LET Interval shall be described by a [TimingDescriptionEventChain](#). `category==SL_LET_INTERVAL`」

The `TimingDescriptionEventChain.category==SL_LET_INTERVAL` shall reference two [TimingDescriptionEvents](#) to represent the `stimulus` (see [\[TPS_TIMEX_00111\]](#)) and the `response` (see [\[TPS_TIMEX_00114\]](#)).

[TPS_TIMEX_00115] Representation of an SL-LET interval release

Status: DRAFT

Upstream requirements: [RS_TIMEX_00022](#)

「An SL-LET Interval release shall be described by `stimulus` referencing a [TDE-ventVfb](#)」

[TPS_TIMEX_00116] Representation of an SL-LET interval terminate

Status: DRAFT

 Upstream requirements: [RS_TIMEX_00022](#)

[An SL-LET Interval terminate shall be described by [response](#) referencing a [TDE-ventVfb](#)]

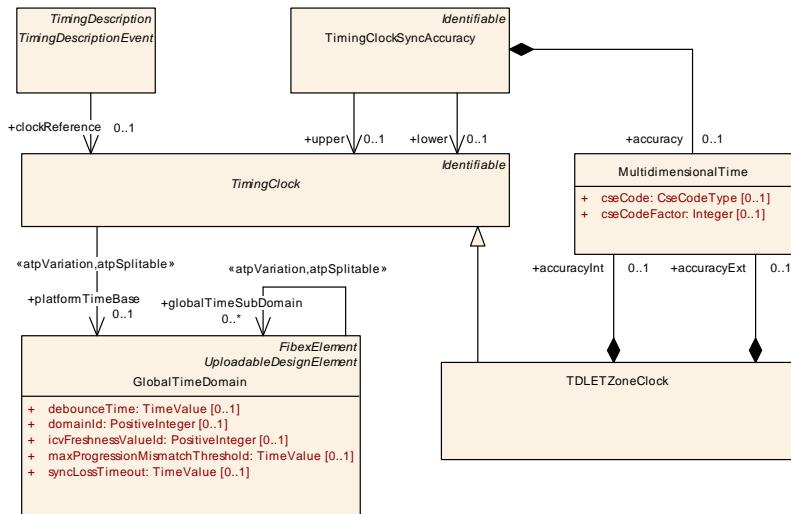


Figure 3.12: TDLETZoneClock and TimingClockSyncAccuracy

The attribute `TimingDescriptionEventChain.isPipeliningPermitted` is provided to specify whether the `TimingDescriptionEventChain.category==SL-LET_INTERVAL` must use a pipelined execution or not [10].

[constr_6816] Restricted usage of `TimingDescriptionEventChain.isPipeliningPermitted` in `TimingDescriptionEventChain`

Status: DRAFT

 Imposition time: [IT_SwcTd](#)

[The attribute `isPipeliningPermitted` shall only exist if the `TimingDescriptionEventChain.category==SL LET INTERVAL`.]

A `TimingDescriptionEvent` used in this context may reference an optional `TimingDescriptionEvent.clockReference` (3.2.2) but shall be locked out for other contexts other than SL-LET.

[constr_6817] Restricted usage of `TimingDescriptionEvent.clockReference`

Status: DRAFT

 Imposition time: [IT_SwcTd](#)

[The reference `TimingDescriptionEvent.clockReference` shall exist if (and only if), the `TimingDescriptionEvent` is itself referenced:

- in the role `stimulus` or

- in the role `response`

by a `TimingDescriptionEventChain.category==SL LET INTERVAL`

The listing in [C.12](#) shows an ARXML which specifies the timing description event chain and timing description events. Pictorially, figure [3.13](#) shows a simplified model of an SL-LET interval.

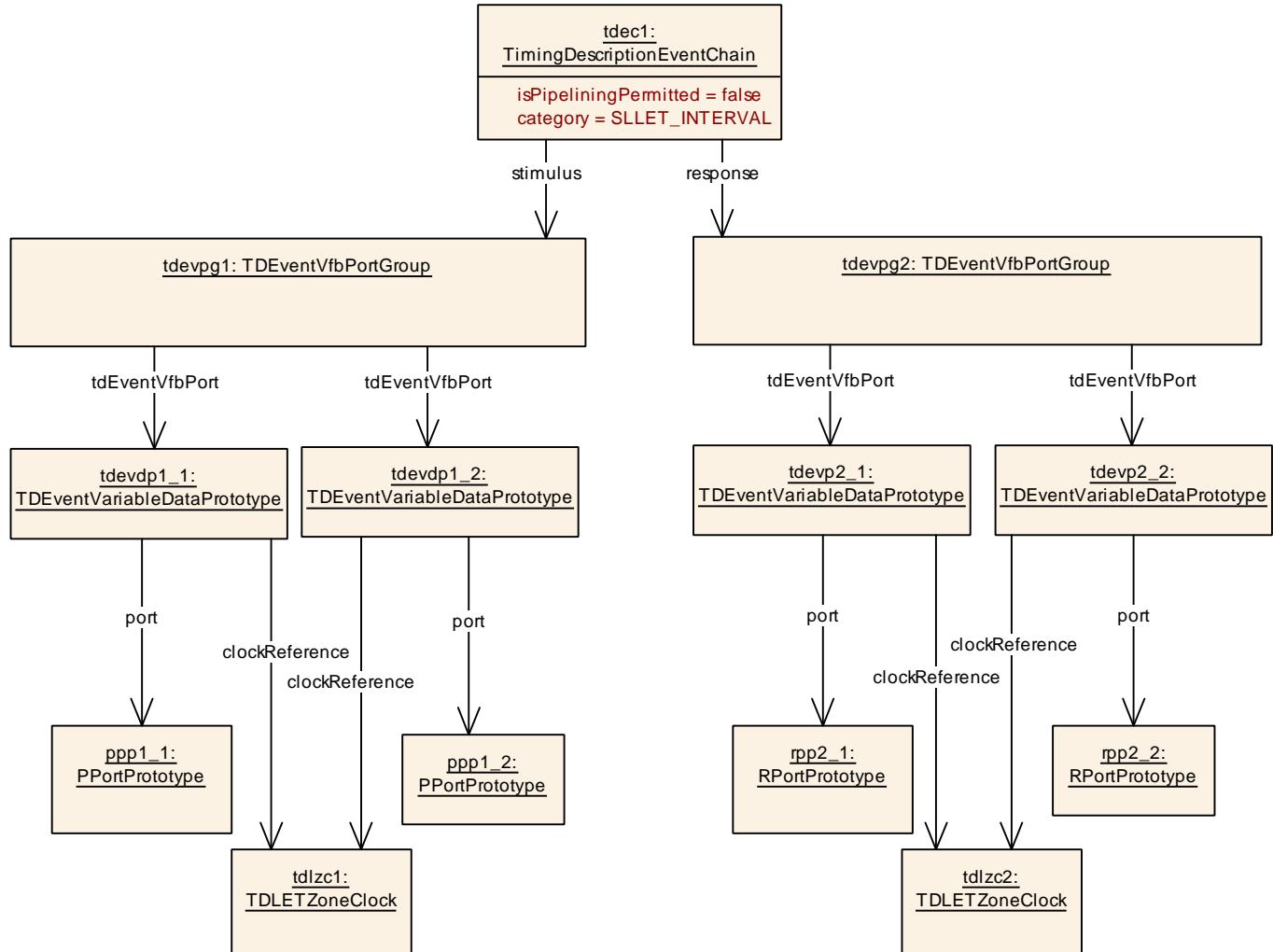


Figure 3.13: SL-LET Interval Model

The timing parameters of an SL-LET interval are specified in the same fashion as an LET interval and shall be specified according to the explanation in [3.1.1](#). Example ARXMLs are shown in [C.12](#) (duration) and [C.12](#) (recurrence).

3.2.1.1 Execution Order

Figure [3.14](#) shows an example regarding *execution order*. The release and terminate event of the SL-LET interval $SLLET_1$ is related to the required port of $Swc01$ and

the provided port of *Swc02*. This implies the execution order $R_1 \rightarrow R_2 \rightarrow R_3$ for the runnable entities. The example is platform agnostic.

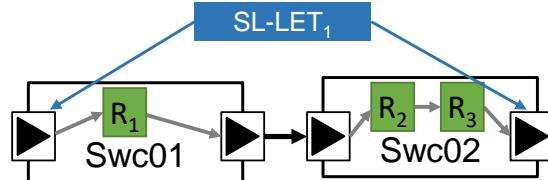


Figure 3.14: Execution order with SL-LET events

3.2.2 SL-LET Time Zones

([LET Zones](#)) are an extension within the domain of [SL-LET](#). The background is that timing description events in a timing event chain *may* have actual different physical time bases ([LET Zone Clocks](#)) and therefore need to reference those [LET Zone Clocks](#) discretely to determine synchronization accuracy for a SL-LET interval.

In a deployment view [LET Zone Clocks](#) may be structured in a tree-based hierarchy. As the tree is descended from a high-level to a low-level view, thus the time bases ([LET Zone Clock](#))s move from a logical high-level e.g. vehicle-level to a e.g. logical low-level sensor/actuator-level.

Figure [3.15](#) shows a graphical example of these concepts. In this example 3 LET-zones are used to represent the hierarchy, in principal any number of LET Zones could be possible but the synchronization accuracy is always between the upper and lower LET zones (timing description event end points).

A LET-zone called TZ_A is represented by a [TDLETZoneClock](#). By referencing the corresponding [TimingClocks](#) within the [TimingDescriptionEvents](#), a SL-LET interval can either be specified between the LET zones TZ_1 and TZ_2 or on a higher hierarchical level with respect to LET zone TZ_0 .

The LET-zone TZ_B is a lower LET-zone of TZ_A , if there exists a [TimingClockSyncAccuracy.upper](#) referring to a [TDLETZoneClock](#) A and a [lower](#) pointing to [TDLETZoneClock](#) B. This definition can be applied recursively, so if TZ_C is a lower LET-zone of TZ_B and TZ_B is a lower LET-zone of TZ_A , then TZ_C is also a lower LET-zone of TZ_A .

The internal synchronization accuracy requirement $\epsilon_{0,int}$ (attribute [accuracyInt](#) of [TDLETZoneClock](#)) is therefore used as a wrapper to abstract lower LET-zones of TZ_0 .

Different LET zone clocks (which can be specified by [TDLETZoneClock](#)) are synchronized in a hierarchical manner. This relation can be specified by the element [TimingClockSyncAccuracy](#). Figure [3.12](#) shows the relation between the elements.

The corresponding ARXML specification is shown in listing [C.12](#).

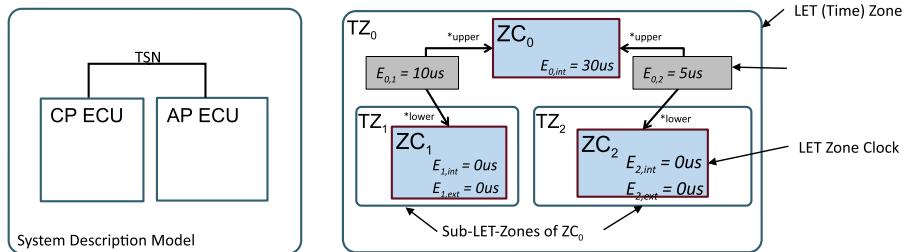


Figure 3.15: Exemplary definition of three LET Zones within a vehicle

In Figure 3.15, the internal synchronization accuracy requirement of $\epsilon_{0,int} = 30\mu s$, which is given as an attribute of [TDLETZoneClock](#), is fulfilled, since the worst-case between any two lower LET-zones of TZ_0 is $\epsilon_{1,ext} + \epsilon_{0,1} + \epsilon_{0,2} + \epsilon_{2,ext} = 0\mu s + 10\mu s + 5\mu s + 0\mu s = 15\mu s \leq 30\mu s$.

Figure 3.16 shows an example when the LET zone TZ_2 shall include further lower LET zones. At the end, the SL-LET interval SL-LET-1 will reference the zone clock ZC_1 and the - not yet defined - zone clock ZC_x . Such a hierarchical concretization can be constrained in advance, by specifying the attributes [accuracyExt](#) and [accuracyInt](#) of [TDLETZoneClock](#).

Without knowledge of the specific lower LET zones, a SL-LET interval SL-LET-1 can be specified with respect to ZC_2 . If both, the release as well as the terminate event of the SL-LET interval relate to ZC_2 , the internal synchronization accuracy [accuracyInt](#) of ZC_2 has to be considered. This abstracts the synchronization between any two lower LET-zones of TZ_2 . If only one of the two events relates to ZC_2 and the other one for example relates to ZC_1 , the external synchronization accuracy [accuracyExt](#) of ZC_2 has to be considered. This abstracts the synchronization accuracy between a lower LET-zone of TZ_2 and any foreign LET-zone.

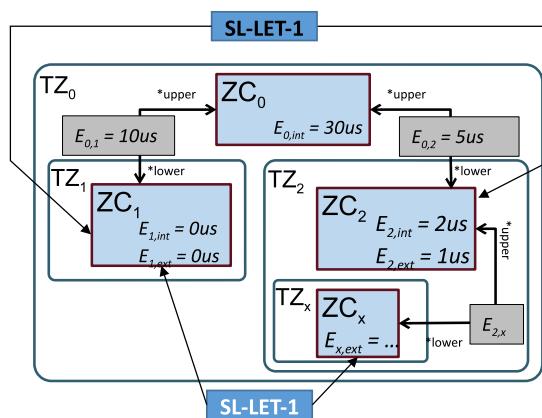


Figure 3.16: Exemplary definition of hierarchical LET Zones

3.2.2.1 SL-LET Zone Clocks

This section details the [TimingClock](#).

[TPS_TIMEX_00100] Optionality of `accuracyInt`

Status: DRAFT

Upstream requirements: [RS_TIMEX_00026](#)

〔If the attribute `accuracyInt` is unspecified, it shall be treated as zero. This is the case, if a LET-zone has no lower LET-zones.〕

[TPS_TIMEX_00103] Optionality of `accuracyExt`

Status: DRAFT

Upstream requirements: [RS_TIMEX_00026](#)

〔If the attribute `accuracyExt` is unspecified, it shall be treated as zero. This is the case, if a LET-zone has no lower LET-zones.〕

3.2.2.2 SL-LET Zone Clock Accuracy

[TPS_TIMEX_00108] Usage of the attribute `accuracy` of `TimingClockSyncAccuracy`

Status: DRAFT

Upstream requirements: [RS_TIMEX_00026](#)

〔The attribute `accuracy` of `TimingClockSyncAccuracy` specifies the synchronization accuracy between the higher level `TDLETZoneClock`, which is referenced by the `upper` attribute and a lower level `TDLETZoneClock`, which is referenced by the `lower` attribute.〕

3.3 Classic Platform Software Clusters

This section describes the support for specifying timing models for *software clusters*. The notion of software cluster and its purpose are described in more detail in [3].

During system design *software clusters* are defined. Software clusters consist of an arbitrary number of software components as sketched in Figure 3.17. The internal structure of those software clusters, like further composition software components and atomic software components, are not known at system design time. Typically, those internals, especially the internal behaviors of the atomic software components along with runnable entities, are the result of further steps taken during the development of the software components. Eventually, the runnable entities need to be assigned to appropriate operating system tasks to ensure the proper execution and therefore a correct provision of the expected functionality realized by the runnable entities. Or, in other words, the operating system being used must provide the needed execution contexts for the runnable entities of a software cluster, if the software cluster is supposed to provide such temporal and dynamic resources.

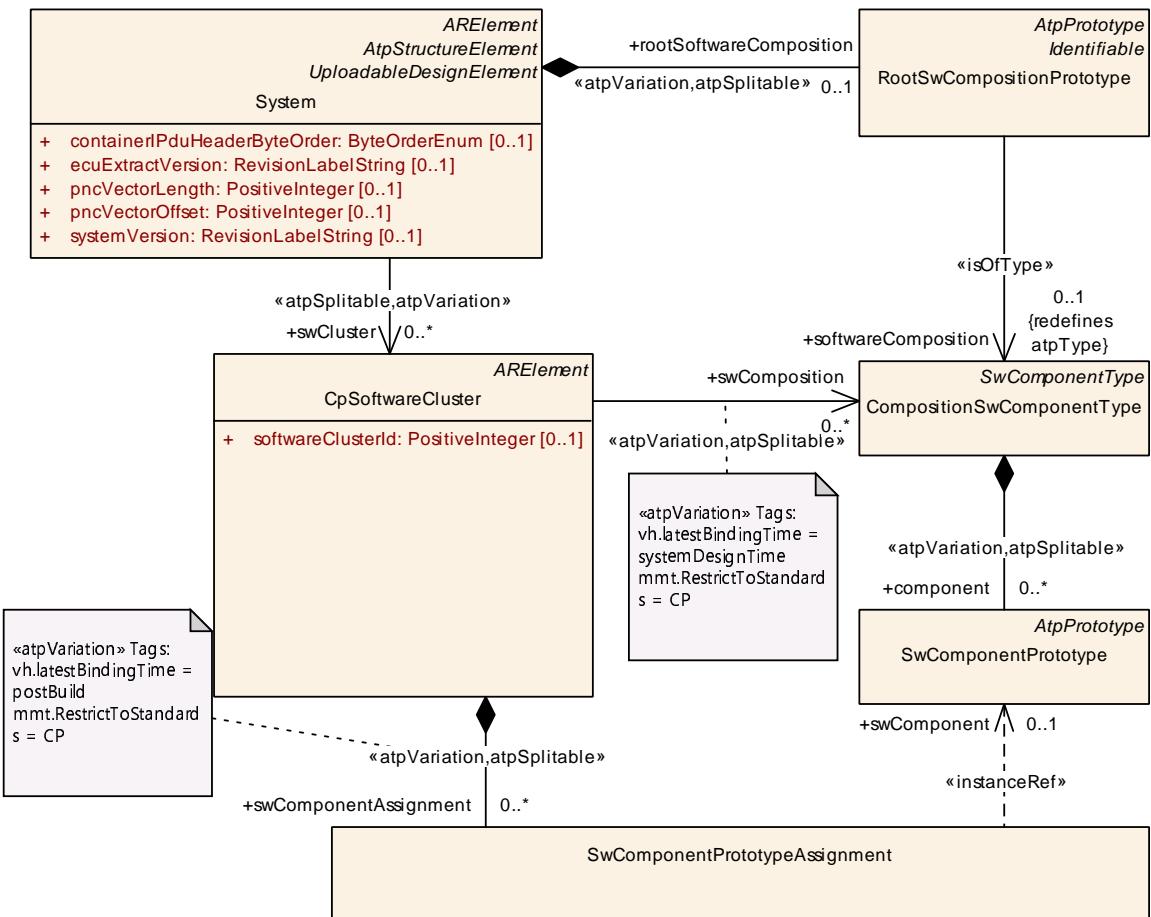


Figure 3.17: Software Cluster Overview

In order to provide a frame at the beginning and during the system design, the elements available in TIMEX are used for describing the temporal and dynamic requirements and characteristics imposed on the execution environment that is needed for proper execution of the software cluster's runnable entities. The notion behind the described approach is that the execution contexts needed by possible runnable entities of a software cluster are abstracted using the elements [TimingDescriptionEvent](#), [TimingDescriptionEventChain](#), and [TimingConstraint](#) to specify the temporal parameters of such execution contexts. The execution context is called *dispatcher* as an abstraction for a system event that occurs to activate the execution of a set of runnable entities in the target system.

The way the AUTOSAR elements are utilized is described in the remaining part of this section.

Since software clusters are defined during system design respectively on the AUTOSAR software system level, the system timing view ([SystemTiming](#)) is used for creating the corresponding timing models. This is shown in the Figure 3.18.

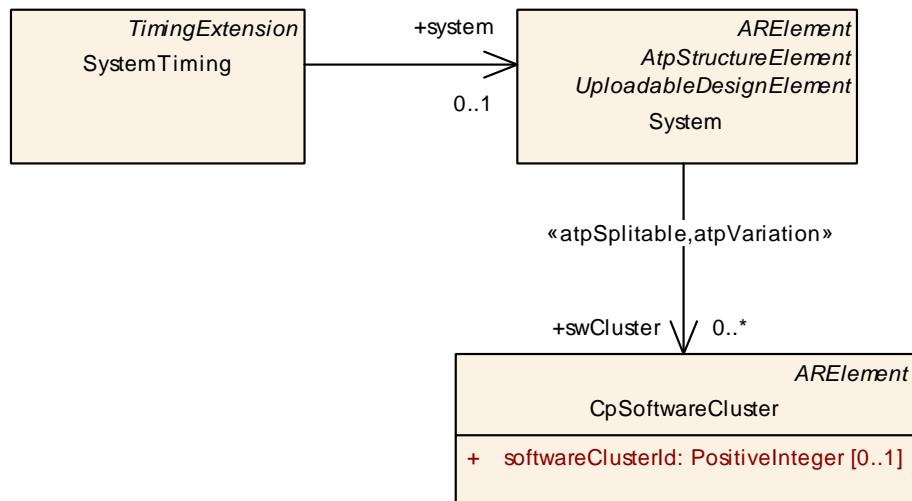


Figure 3.18: System timing used to specify the timing for one or more software clusters

A system timing describing the timing model for a software cluster or a group of software clusters references a system that in turn references one or more software clusters. In the latter case, the scope of the timing model is the set of referenced software clusters. Primarily, the timing model of a system timing for a software cluster contains the definition of the execution context – dispatchers – required for the proper execution of the runnable entities that are later defined in the development.

[TPS_TIMEX_00127] *SystemTiming* of classic platform software clusters

Status: DRAFT

[A *SystemTiming* which references a *System.category==SW_CLUSTER_SYSTEM_DESCRIPTION* ([TPS_SYST_01003]) is intended to observe the timing of Classic Platform software clusters.]

3.3.1 Specifying a Dispatcher

The value of the attribute *category* of a *TimingDescriptionEvent* indicates if the timing description represents the entry point or the exit point point of a dispatcher. The value *DISPATCH_ENTRY_POINT* indicates the point in time the dispatcher is entered. And the value *DISPATCH_EXIT_POINT* indicates the point in time the dispatcher is exited.

[constr_4561] Usage of the category value *DISPATCH_ENTRY_POINT* in *TimingDescriptionEvent*

Imposition time: *IT_SysTd*

[The value *DISPATCH_ENTRY_POINT* of the attribute *category* of a *TimingDescriptionEvent* shall be set if and only if the timing description event plays the role of a stimulus event and the corresponding timing description event chain, referencing

this timing description event, represents a dispatcher in the context of describing timing of a software cluster.]

[constr_4562] Usage of the category value `DISPATCH_EXIT_POINT` in `TimingDescriptionEvent`

Imposition time: IT_SysTd

[The value `DISPATCH_EXIT_POINT` of the attribute `category` of a `TimingDescriptionEvent` shall be set if and only if the timing description event plays the role of a response event and the corresponding timing description event chain, referencing this timing description event, represents a dispatcher in the context of describing timing of a software cluster.]

As sketched in Figure 3.19 a dispatcher for a software cluster is specified by a timing description event chain `TimingDescriptionEventChain`. The timing description event chain references two timing description events `TimingDescriptionEvents` representing the dispatcher's entry point (stimulus event of the timing description event chain) and the dispatcher's exit point (response event of the timing description event chain). The value of the first element's attribute `category` is `DISPATCH_ENTRY_POINT`; and the value of the latter element's attribute `category` is `DISPATCH_EXIT_POINT`. The complex timing event `TDEventComplex` is used for the purpose to indicate the event that occurs when the dispatcher is entered and exited. The complex timing event is used because both timing description events are not referencing an *observable* location, like the other types of timing description events. The event occurrence expression of the complex timing event may be used to specify the particular event supposed to occur in case of entering and exiting the dispatcher.

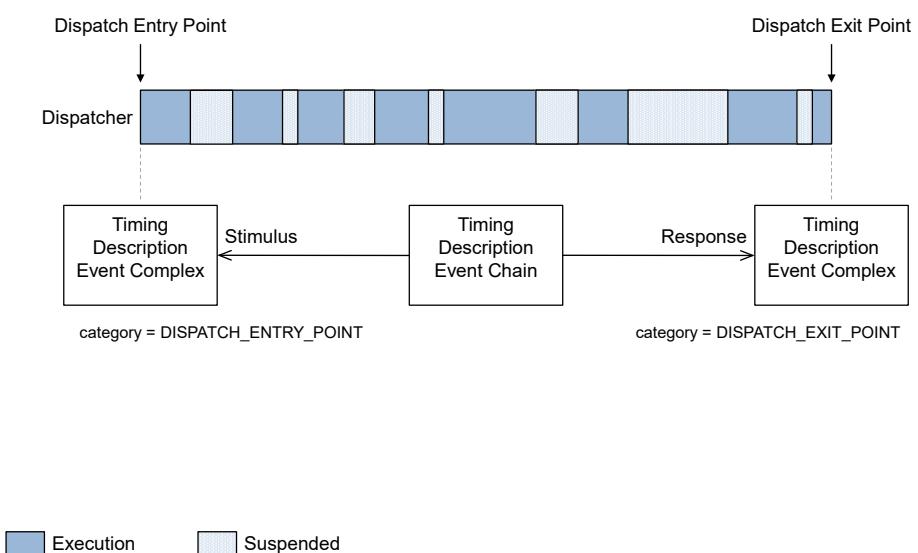


Figure 3.19: Specifying a Dispatcher for a Software Cluster

The listing C.11 shows an ARXML fragment that matches what is shown in Figure 3.19. It specifies a timing description event chain, and two timing description events representing the entry and exit point of this dispatcher. The former timing description event

plays the role *stimulus* and the latter timing description event plays the role *response* of the timing description event chain representing the dispatcher.

The attribute `category` of a `TimingDescriptionEvent` indicates the use of this element in the context of timing models for software clusters respectively dispatchers.

The value of the attribute `category` of a `TimingDescriptionEvent` indicates if the timing description event represents the dispatch entry point of a dispatcher belonging to a software cluster or the dispatch exit point of a dispatcher belonging to a software cluster. The value `DISPATCH_ENTRY_POINT` indicates the point in time the dispatcher is entered respectively activated. And the value `DISPATCH_EXIT_POINT` indicates the point in time the dispatcher is exited respectively terminated. For more details refer to section 2.3.2 and in particular constraints [\[constr_4559\]](#), [\[constr_4561\]](#) and [\[constr_4562\]](#).

3.3.2 Specifying Timing Parameters of a Dispatcher

The timing parameters of a dispatcher are specified in the same fashion as an LET interval and shall be specified according to the explanation in 3.1 particularly the *maximum execution time (duration)* of a dispatcher.

A latency timing constraint is imposed on a timing description event chain representing a dispatcher and the attribute `maximum` specifies the maximum execution time of the dispatcher, as shown in Figure 3.20. The value of the attribute `minimum` shall be set to zero (0) because it is not expected that a dispatcher *must* execute for a minimum period of time.

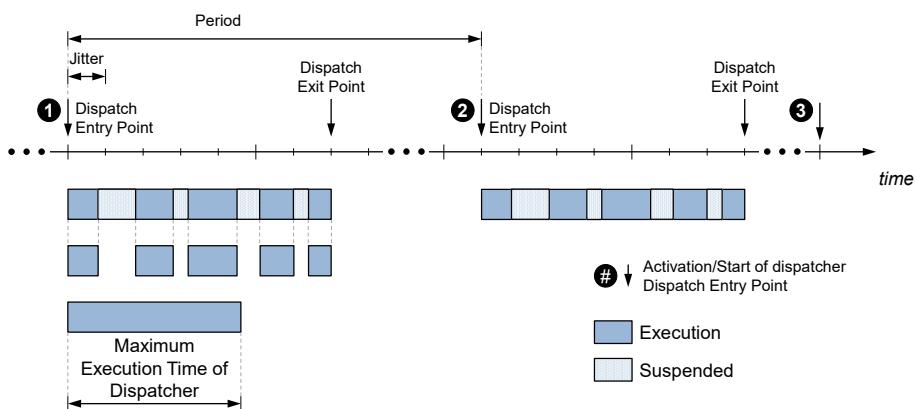


Figure 3.20: Maximum execution time of a dispatcher

As depicted in Figure 3.20 the timing parameters of a dispatcher are specified by a latency timing constraint and periodic event triggering constraint. The latency timing constraint specifies the maximum execution time of a dispatcher within the given period specified by the periodic event triggering constraint.

The listing C.11 shows an ARXML fragment which defines a latency timing constraint of $500\mu\text{s}$ that is imposed on the timing description event chain — dispatcher — defined in

listing C.11. In essence, this latency timing constraint specifies the maximum execution time of the dispatcher to be $500\mu\text{s}$.

The recurrence of a dispatcher is specified using the periodic event triggering constraint [PeriodicEventTriggering](#). This timing constraint is imposed on the stimulus event — the entry point of the dispatcher — of the timing description event chain representing a dispatcher. The listing C.11 shows an ARXML fragment which defines a periodic event triggering constraint that specifies a recurrence of 10ms; in other words the dispatcher is entered every 10ms. Note that the execution of a dispatcher is supposed to be completed before the dispatcher is started — entered — again.

3.3.3 Mapping Executable Entities to a Dispatcher

Eventually, the executable entities belonging to the internal behavior of atomic software components within the software clusters shall be assigned to temporal and dynamic resources that are provided by the available *host* software cluster. Since the temporal resource is either a dispatcher or a LET interval, the executable entities need to be assigned to either of them.

In both cases the capabilities of the element [ExecutionOrderConstraint](#) are used for this purpose. To assign executable entities to LET intervals is described in section 3.1.3 in detail.

An [ExecutionOrderConstraint](#) is specified consisting of [EOCEventRef](#)s which reference the RTE events of the executable entities that shall be assigned to the dispatcher. These [EOCEventRef](#)s are specified as nested elements in an [EOCExecutableEntityRefGroup](#). The [triggeringEvent](#) of the [EOCExecutableEntityRefGroup](#) references the [TDEventComplex](#) that plays the role of a dispatch entry point respectively dispatcher.

The listing C.11 shows an ARXML fragment which specifies that the runnable entity one (1) of application software component five (5), runnable entity one (1) of application software component two (2), runnable entity two (2) of application software component 15, runnable entity two (2) of application software component eight (8), runnable entity one (1) of application software component 13, and runnable entity three (3) of application software component nine (9) are executed in the context of the dispatcher called "Dispatcher10ms". Those runnable entities are all part of application software components that are part of the application software cluster called "SoftwareClusterA".

3.3.4 Mappings between Software Clusters

An important step during the development of AUTOSAR software systems is to specify contracts between various parties that are involved in the development. To specify provided and needed temporal and dynamic resources the *software cluster mappings* are used. The purpose of these mappings are firstly to unequivocally identify a temporal and dynamic resource, and secondly to specify the software cluster providing such

resources and specify the software clusters that request such resources. The Figure 3.21 shows these mapping elements and their relationships.

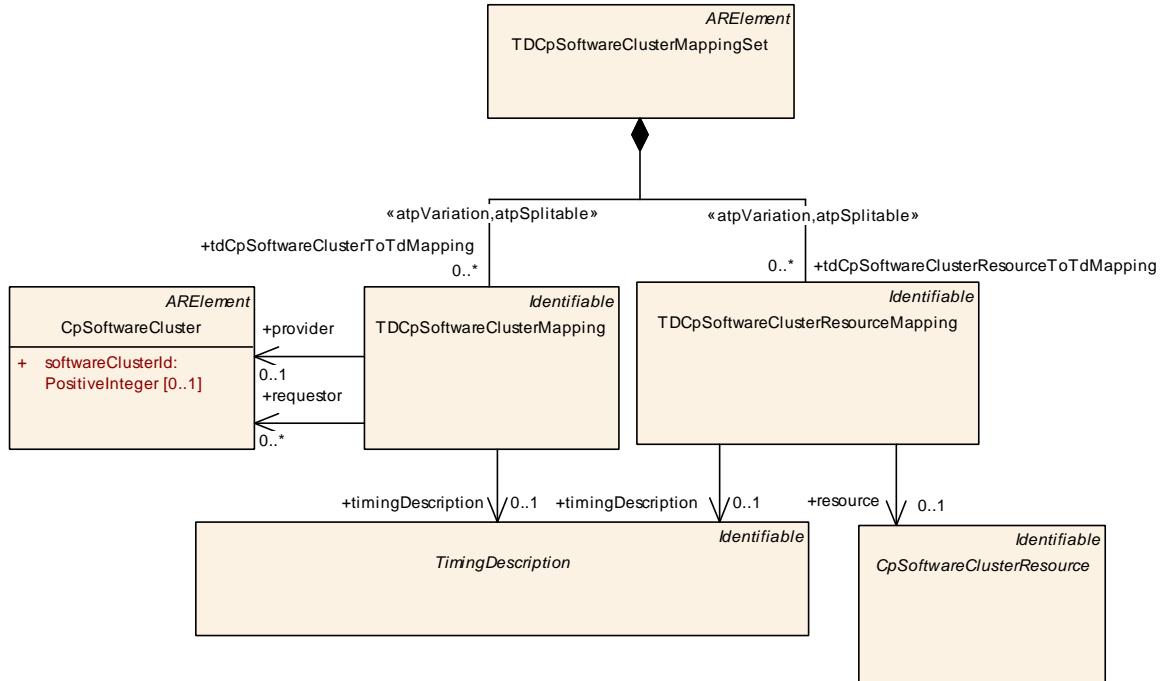


Figure 3.21: TimingCpSoftwareClusterMapping

| | | | | |
|--|---|--------------|-------------|---|
| Class | TDCpSoftwareClusterMappingSet | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingCpSoftwareCluster | | | |
| Note | This is used to gather of classic platform software cluster mappings. Tags: atp.recommendedPackage=TimingExtensions | | | |
| Base | <i>ARElement, AROObject, CollectableElement, Identifiable, MultilanguageReferrable, Packageable Element, Referrable</i> | | | |
| Aggregated by | ARPackage.element | | | |
| Attribute | Type | Mult. | Kind | Note |
| tdCpSoftwareClusterResourceToTdMapping | TDCpSoftwareClusterResourceMapping | * | aggr | Maps a CP software cluster resource to a temporal resource. Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=tdCpSoftwareClusterResourceToTdMapping.shortName, tdCpSoftwareClusterResourceToTdMapping.variationPoint.shortLabel vh.latestBindingTime=postBuild |
| tdCpSoftwareClusterToTdMapping | TDCpSoftwareClusterMapping | * | aggr | Maps a temporal resource to a mapping between a providing CP software cluster and requesting CP software clusters. Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=tdCpSoftwareClusterToTdMapping.shortName, tdCpSoftwareClusterToTdMapping.variationPoint.shortLabel vh.latestBindingTime=postBuild |

Table 3.5: TDCpSoftwareClusterMappingSet

| Class | TDCpSoftwareClusterMapping | | | |
|--------------------|--|-------|------|---|
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingCpSoftwareCluster | | | |
| Note | This is used to specify a mapping between a software cluster that provides temporal and dynamic resources and the software clusters that need these resources. | | | |
| Base | ARObject, Identifiable, MultilanguageReferable, Referrable | | | |
| Aggregated by | TDCpSoftwareClusterMappingSet.tdCpSoftwareClusterToTdMapping | | | |
| Attribute | Type | Mult. | Kind | Note |
| provider | CpSoftwareCluster | 0..1 | ref | This is the software cluster that provides the temporal and dynamic resource. |
| requestor | CpSoftwareCluster | * | ref | This is the software cluster that requests the temporal and dynamic resource. |
| timing Description | TimingDescription | 0..1 | ref | The timing description representing the temporal and dynamic resource. |

Table 3.6: TDCpSoftwareClusterMapping

[constr_6918] Referenced TimingDescriptions in TDCpSoftwareClusterMapping and TDCpSoftwareClusterResourceMapping

Status: DRAFT

Imposition time: IT_SysTd

〔The references:

- TDCpSoftwareClusterMapping.timingDescription
- TDCpSoftwareClusterResourceMapping.timingDescription

shall refer to either:

- TDEventComplex.category==DISPATCH_ENTRY_POINT, or
- TimingDescriptionEventChain.category==LET_INTERVAL

〕

[constr_6919] Referenced CpSoftwareCluster of TDCpSoftwareClusterMapping

Status: DRAFT

Imposition time: IT_SysTd

〔The references:

- TDCpSoftwareClusterMapping.provider
- TDCpSoftwareClusterMapping.requestor

shall refer to a CpSoftwareCluster.category==HOST_SOFTWARE_CLUSTER〕

| Class | TDCpSoftwareClusterResourceMapping | | | |
|--------------------|--|-------|------|---|
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingCpSoftwareCluster | | | |
| Note | This is used to assign an unequivocal global resource identification to a temporal and dynamic resource. | | | |
| Base | ARObject, <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i> | | | |
| Aggregated by | TDCpSoftwareClusterMappingSet.tdCpSoftwareClusterResourceToTdMapping | | | |
| Attribute | Type | Mult. | Kind | Note |
| resource | CpSoftwareCluster Resource | 0..1 | ref | The specific resource identification assigned to the temporal and dynamic resource. |
| timing Description | TimingDescription | 0..1 | ref | The timing description representing the temporal and dynamic resource. |

Table 3.7: TDCpSoftwareClusterResourceMapping

[constr_4565] Consistency of `TDCpSoftwareClusterMapping.timingDescription` and `TDCpSoftwareClusterResourceMapping.timingDescription`

Imposition time: IT_SysTd

「The references:

- `TDCpSoftwareClusterMapping.timingDescription`
- `TDCpSoftwareClusterResourceMapping.timingDescription`

shall (after [constr_6918] has been applied) refer to the same sub-class and *category of TimingDescription*」

The listing C.11 shows an ARXML fragment which describes that the *application* software cluster called "SoftwareClusterA" requires two temporal resources provided by the *host* software cluster called "SoftwareClusterHost". The two required temporal resources are the dispatcher called "Dispatcher10ms" and the logical execution time interval called "Let5msInterval".

3.3.5 Example

In the given example, an AUTOSAR software system consists of eleven (11) composition software components. The composition software component *type* called "CSWCT01" plays the role of the root software composition software prototype of this system. The composition software component type contains two composition software component *prototypes* called "cSwcPt02" and "cSwcPt03". The composition software component type "CSWCPT02" contains two composition software component prototypes called "cSwcPt04" and "cSwcPt09"; and the composition software component type called "CSWCT03" contains the composition software component prototypes called "cSwcPt05 and cSwcPt06". The composition software component type called "CSWCT04" contains two composition software component prototypes called "cSwcPt07" and "cSwcPt08". And last but not least, the composition software component type called "CSWCT06" contains two composition software component prototypes

called "cSwcPt10" and "cSwcPt11". The composition hierarchy of the software components is shown in Figure 3.22.

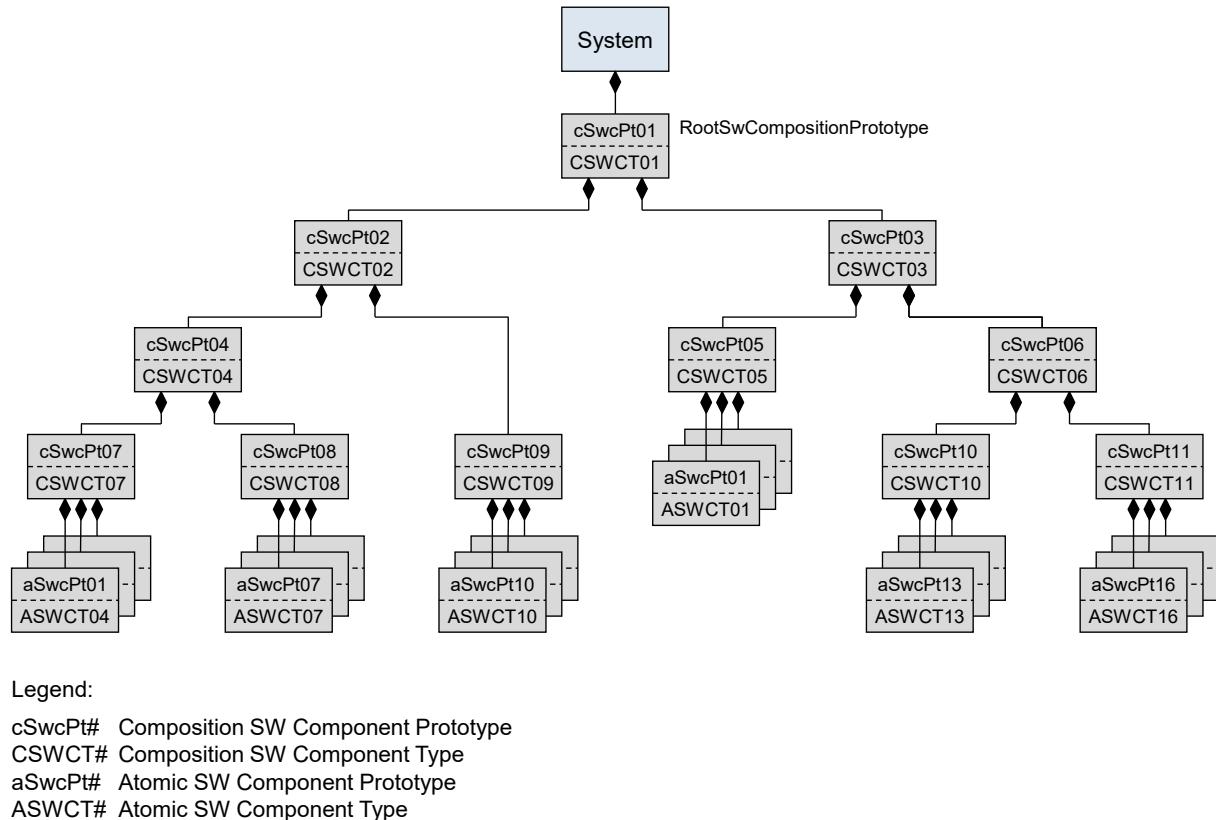


Figure 3.22: Example Software Cluster

Each of the composition software component types called "CSWCT05", "CSWCT07", "CSWCT08", "CSWCT09", "CSWCT10" and "CSWCT11" contain three atomic software component prototypes. The names of these atomic software component prototypes are following the rule "aSwcPt##". The placeholder "##", representing a two digit number with leading zero, ranges from 01 to 18. For example, the composition software component type "CSWCT05" contains the three atomic software component prototypes called "aSwcPt01", "aSwcPt02" and "aSwcPt03". Any of the atomic software component types have an internal behavior and each of those internal behaviors consists of three runnable entities named "re#ASWCT##"; for example, the name "re1ASWCT05" denotes the first runnable entity of the atomic software component type five (5). An RTE Event — Timing Event — is specified for each of the runnable entities, and is named as follows: "teRe#ASWCT##".

The described software system consists of two software clusters called "Software Cluster A" and "Software Cluster B". This is shown in the Figure 3.23. The software cluster A consists of the composition software components "cSwcPt04", "cSwcPt05", "cSwcPt07", "cSwcPt08" and "cSwcPt10". And the software cluster B consists of the composition software components "cSwcPt09" and "cSwcPt11".

Legend:

cSwcPt# Composition SW Component Prototype

CSWCT# Composition SW Component Type

aSwcPt# Atomic SW Component Prototype

ASWCT# Atomic SW Component Type

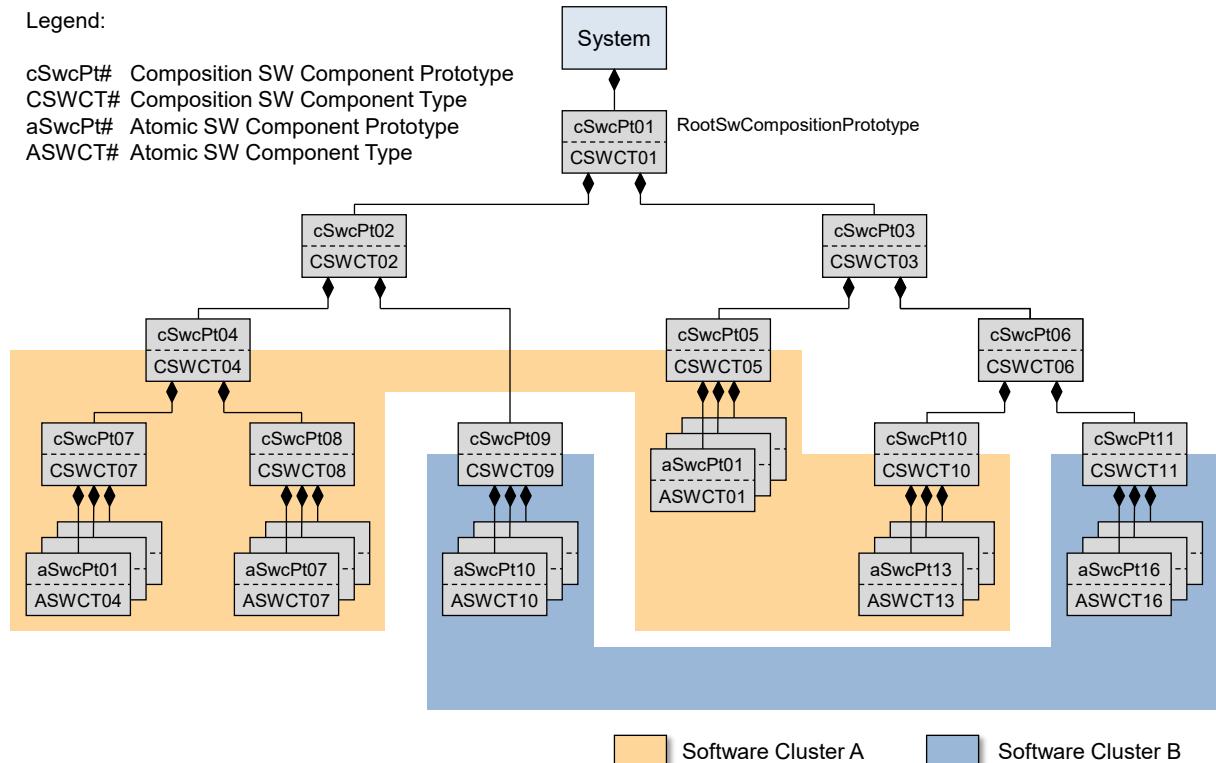


Figure 3.23: System consisting of two Software Clusters

The assumption is that the temporal resources, needed by the software cluster A and software cluster B, are provided by a third software cluster representing the *host* software cluster.

A Reference Material

A.1 Terms and Abbreviations

The main list of terms and abbreviations are defined in [7]. The following table contains the list of terms and abbreviations used in the scope of this document which are not already defined in [7] along with the spelled-out meaning of each of the abbreviations.

| Abbreviation | Meaning |
|--------------|----------------------------|
| EOC | Execution Order Constraint |
| TD | Timing Description |

Table A.1: Abbreviations used in the scope of this Document

| Term | Meaning |
|---------------------------|--|
| Jitter | For a periodically occurring timing event, the jitter is defined as the maximum variation of its period with respect to a predefined standard period. |
| Latency | The latency of a timing event chain describes the time duration between the occurrence of the stimulus and the occurrence of the corresponding response. |
| Maximum interarrival time | Describes the maximum time interval between two consecutive event occurrences. In the more general case, this attribute is an array of the maximum latency between two, three, four, ... event occurrences. |
| Minimum interarrival time | Describes the minimum time interval between two consecutive event occurrences. In the more general case, this attribute is an array of the minimum latency between two, three, four, ... event occurrences. |
| Period | Describes the expected time interval between two consecutive event occurrences, neglecting variation (jitter). |
| Synchronization | Synchronization focuses on the occurrence of different timing events. Synchronization of timing events means that they shall occur simultaneously within a certain tolerance interval. |
| Timing analysis | Timing analysis is a method of determining the timing behavior of the system. This includes consideration of timing relevant system behavior like task preemptions, interrupt handling, resource blocking, etc. |
| Timing path | A timing path defines a sequence of communication or computation activities of the system, whose timing behavior shall be examined. Timing paths can be expressed by event chains. |
| Timing property | A timing property defines the state or value of a timing relevant aspect within the system. Thus, a property does not represent a constraint for the system, but a somehow gathered (e.g. measured, estimated or determined) or defined attribute of the system. |
| Timing requirement | A timing requirement defines a restriction on timing that shall be fulfilled to ensure proper operation of the system. Timing requirements can be expressed by using timing constraints. |

Table A.2: Terms used in the scope of this Document

A.2 Imposition Times

| Imposition Time | Description | Motivation |
|-------------------|--|--|
| IT_BswTd | Bsw Timing Description is completed | This imposition time is aimed at the time when a Bsw Timing is complete. This applies for both the Bsw Module Timing and the Bsw Composition Timing. |
| IT_EcuTd | Ecu Timing Description is completed | This imposition time is aimed at the time when a Ecu-wide Timing is complete. |
| IT_SubClasTdEv | Imposition time associated with the concrete subclass of Timing DescriptionEvent by condition. | The imposition time is associated with the concrete subclass of TimingDescriptionEvent if the constraint is applied to a TimingDescriptionEvent or at the imposition time associated with the concrete subclass of TimingExtension if the constraint is applied to a TimingDescriptionEventChain . |
| IT_SubClasTdEvAss | Imposition time associated with the concrete subclass of Timing DescriptionEvent is applied. | This means that the imposition time of the constraint cannot be unambiguously defined on the level of the abstract meta-class TimingDescriptionEvent . Sub-classes of TimingDescriptionEvent have imposition times associated with them (by means of constraints that refer to the subclasses) and the constraints that apply in the context of the definition of TimingDescriptionEvent shall therefore not contain a concrete imposition time but take over the imposition time from the applicable subclass. Example: subclass TDEventVfb is associated with the imposition time at the time when the VFB Timing Description is complete. |
| IT_SubClasTeAss | Imposition time associated with the concrete subclass of Timing Extension. | This means that the imposition time is relative to the concrete subclass of TimingExtension (Timing View) in use, namely: <ul style="list-style-type: none"> - at the time when the VFB Timing Description is complete, - at the time when the Swc Timing Description is complete, - at the time when the System Timing Description is complete, - at the time when the Bsw Timing Description is complete, - at the time when the Ecu Timing Description is complete |
| IT_SwcTd | Swc Timing Description is completed | This imposition time is aimed at the time when a Swc Timing is complete. |
| IT_SysTd | System Timing Description is completed | This imposition time is aimed at the time when a System Timing is complete. |
| IT_VfbTd | VFB Timing Description is completed | This imposition time is aimed at the time when a VFB Timing is complete. |

Table A.3: Imposition Times of constraints in this document

A.3 Requirements Traceability

The following table references the requirements specified in AUTOSAR RS Timing Extensions [11] and denotes how each of them are satisfied by the meta-model.

| Requirement | Description | Satisfied by |
|------------------|--|---|
| [RS_TIMEX_00001] | Timing properties | [TPS_TIMEX_00001] [TPS_TIMEX_00002] [TPS_TIMEX_00003] [TPS_TIMEX_00004] [TPS_TIMEX_00006] [TPS_TIMEX_00007] [TPS_TIMEX_00008] [TPS_TIMEX_00010] [TPS_TIMEX_00011] [TPS_TIMEX_00012] [TPS_TIMEX_00013] [TPS_TIMEX_00014] [TPS_TIMEX_00015] [TPS_TIMEX_00016] [TPS_TIMEX_00017] [TPS_TIMEX_00018] [TPS_TIMEX_00019] [TPS_TIMEX_00020] [TPS_TIMEX_00021] [TPS_TIMEX_00022] [TPS_TIMEX_00023] [TPS_TIMEX_00024] [TPS_TIMEX_00025] [TPS_TIMEX_00026] [TPS_TIMEX_00027] [TPS_TIMEX_00028] [TPS_TIMEX_00029] [TPS_TIMEX_00030] [TPS_TIMEX_00031] [TPS_TIMEX_00032] [TPS_TIMEX_00033] [TPS_TIMEX_00034] [TPS_TIMEX_00035] [TPS_TIMEX_00036] [TPS_TIMEX_00039] [TPS_TIMEX_00042] [TPS_TIMEX_00043] [TPS_TIMEX_00044] [TPS_TIMEX_00045] [TPS_TIMEX_00052] [TPS_TIMEX_00053] |
| [RS_TIMEX_00002] | Timing constraints | [TPS_TIMEX_00003] [TPS_TIMEX_00004] [TPS_TIMEX_00006] [TPS_TIMEX_00007] [TPS_TIMEX_00010] [TPS_TIMEX_00011] [TPS_TIMEX_00012] [TPS_TIMEX_00013] [TPS_TIMEX_00014] [TPS_TIMEX_00015] |
| [RS_TIMEX_00003] | Optionality of timing constraints | [TPS_TIMEX_00009] |
| [RS_TIMEX_00004] | Event chains | [TPS_TIMEX_00002] |
| [RS_TIMEX_00005] | Structure of event chains | [TPS_TIMEX_00002] |
| [RS_TIMEX_00006] | Triggering behavior of event chains | [TPS_TIMEX_00003] [TPS_TIMEX_00010] [TPS_TIMEX_00011] [TPS_TIMEX_00012] [TPS_TIMEX_00013] [TPS_TIMEX_00014] |
| [RS_TIMEX_00007] | Synchronization of event chains | [TPS_TIMEX_00006] |
| [RS_TIMEX_00008] | Multiple asynchronous time bases | [TPS_TIMEX_00003] [TPS_TIMEX_00006] [TPS_TIMEX_00010] [TPS_TIMEX_00011] [TPS_TIMEX_00012] [TPS_TIMEX_00013] [TPS_TIMEX_00014] [TPS_TIMEX_00015] |
| [RS_TIMEX_00009] | Loop-back signal flow in sender-receiver communication | [TPS_TIMEX_00002] |
| [RS_TIMEX_00011] | Mode dependency | [TPS_TIMEX_00049] [TPS_TIMEX_00050] [TPS_TIMEX_00051] |
| [RS_TIMEX_00012] | Sensor/actuator delay | [TPS_TIMEX_00004] |
| [RS_TIMEX_00013] | Specification of timing resources for software-component description | [TPS_TIMEX_00008] |
| [RS_TIMEX_00014] | Sequence of execution of runnable entities | [TPS_TIMEX_00007] |
| [RS_TIMEX_00015] | Timing-requirements of SW-Components | [TPS_TIMEX_00004] [TPS_TIMEX_00010] |
| [RS_TIMEX_00017] | Synchronization constraint on events | [TPS_TIMEX_00006] |
| [RS_TIMEX_00018] | Predefined events for port interfaces at VFB level | [TPS_TIMEX_00039] |
| [RS_TIMEX_00019] | AUTOSAR Methodology support | [TPS_TIMEX_00020] [TPS_TIMEX_00042] [TPS_TIMEX_00043] [TPS_TIMEX_00044] [TPS_TIMEX_00045] |
| [RS_TIMEX_00020] | Support for events indicating variable accesses | [TPS_TIMEX_00020] [TPS_TIMEX_00044] |
| [RS_TIMEX_00022] | Support for Logical Execution Time | [TPS_TIMEX_00057] [TPS_TIMEX_00112] [TPS_TIMEX_00113] [TPS_TIMEX_00115] [TPS_TIMEX_00116] |
| [RS_TIMEX_00023] | Support for Specifying Synchronization | [TPS_TIMEX_00054] |
| [RS_TIMEX_00026] | Support for System Level Logical Execution Time | [TPS_TIMEX_00100] [TPS_TIMEX_00103] [TPS_TIMEX_00108] [TPS_TIMEX_00125] |

Table A.4: Requirements Tracing

B Test Cases

This chapter outlines two test case examples describing a potential approach to use TIMEX in a practical way. Furthermore, chapter [B.3](#) describes the use of external VFB events in more detail.

B.1 Component integration

One of the main concerns for the usage of the AUTOSAR development methodology and AUTOSAR exchange formats is the need of OEMs and suppliers to exchange specification data in a machine-readable, reliable and straightforward way in order for example to integrate components in systems. The primary purpose of the "Specification of Timing Extensions" is to facilitate requesting a specific timing behavior of such components. And this topic is described in this section in more detail based on an integration scenario.

Integrating a software component instance delivered by an external party requires the provision of timing information related to this component. As this information can be attached to specific [SwComponentType](#), with regards to its communication partners, the according view [VfbTiming](#) (see [2.1.1](#)) is used. Additionally, specific timing constraints for implementing this software component are given, too.

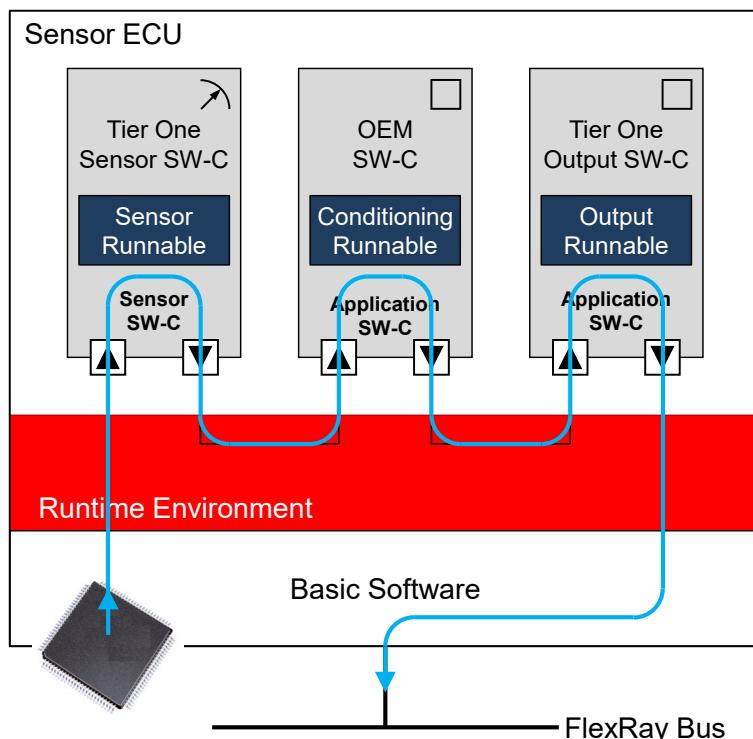


Figure B.1: A Sensor ECU connected with a FlexRay Network and three software components

Figure B.1 outlines the scenario in a demonstrative way. The shown ECU holds three software components: the first one, called "Sensor SW-C", reads data from a hardware sensor; the second one, called "Conditioning SW-C", performs signal data conditioning like filtering and averaging. And last but not least, the third one, called "Output SW-C", converts internal data representations (like 32bit Float) to ready-to-send data representations (like UInt16). As certain requirements for sensor data conditioning as an input for several other functions within the vehicle exist, the software component "Conditioning SW-C" may be delivered by the OEM, directly. A partial description of these components, ports, and interfaces is shown in listing C.16.

In addition to this, receiving the sensor data by other ECUs, which are not shown in the figure, requires this data to fulfill certain timing requirements regarding their maximum age, for example. Mapped to the figure this means that the blue data path drawn shall have a specific temporal length. This requirement is the other hand side of the actual scenario.

The software component "Conditioning SW-C" is delivered by the OEM for the sake of implementing special filtering or averaging algorithms applied on the measured sensor data. Thus, the mapping of software component to this ECU is fixed, already. To fulfill certain non-functional requirements, the implementing `RunnableEntity` of software component "Conditioning" needs to be executed straight away after `RunnableEntity` of component "Sensor SW-C" and right before `RunnableEntity` of component "Output SW-C". In addition, the Tier-1 needs information about the execution times of the runnable entity he has to expect when integrating the software component "Conditioning SW-C". Specifying this can be done by describing the measured (or simulated, estimated, etc.) execution times of the `RunnableEntity`. The following subsections give a brief idea how this can be accomplished by utilizing the capabilities of the AUTOSAR Specification of Timing Extensions.

B.1.1 VFB view

At first, timing descriptions and constraints on VFB level are defined. The component "Conditioning SW-C" receives data via its required port "UnprocessedSensorData" at a specific point in time. This point is denoted by the event "ConditioningReceived", whereas the event "ConditioningSent" denotes the point in time data is sent via the provided port "ProcessedSensorData". To prescribe a "maximum age" for the reading input the `LatencyTimingConstraint` is used. For this, the *external* event "SensorDataProduced" is defined. Based on this, an event chain between this external event "SensorDataProduced" and the "ConditioningReceived" event is specified — the event "SensorDataProduced" plays the role of the stimulus event and the event "ConditioningReceived" is playing the role of the response event with regard to the specified event chain. The latency timing constraint is pointing to this event chain. The representation of the events, event chain and the corresponding timing constraint is shown in listing C.17.

B.1.2 ECU view

After generating the ECU extract, implementation related details of the ECU are available and execution order constraints for the mapped software components — more precise, their runnable entities — exist. For the sake of easiness, each software component implements one [RunnableEntity](#). Constraining their execution order using the [ExecutionOrderConstraint](#) is shown in listing C.18.

Another typical constraint describes the maximum time to be elapsed for sending data on the bus. Therefore, an event "DataTransmitted" representing the point in time the data is sent on the bus is specified using the event type [TDEventFrame](#) (listing C.19). Additionally a [TimingDescriptionEventChain](#) is specified having "ConditioningSent" as stimulus event and "DataTransmitted" as response event (see listing C.20).

The constraint prescribing the maximum latency between the point in time the stimulus event occurs and the point in time the response event occurs is shown in listing C.21.

B.2 Engine control

This example illustrates an example for the definition of timing constraints in an engine management system. Although the system is simplified to be included within this chapter it is based on a real world example in its basic concepts.

B.2.1 Overview

The example system is an air mass controlled gasoline internal combustion engine control system. Roughly, the functionality of software components can be categorized as described in the following:

Sensors Three [SensorActuatorSwComponentType](#)s called "MassAirFlowSensor", "AcceleratorPedalSensor", and "ThrottleSensor" are responsible for reading in the most important control factors.

Application Based Calculation Most [ApplicationSwComponentType](#)s calculate the new control factors for the engine. In summary these components are "AcceleratorPedalVoter", "ThrottleController", "ThrottleChange", "BaseFuelMass", "TransientFuelMass", "Ignition", and "TotalFuelMass".

Actuators The control of the actuators is encapsulated by the [SensorActuator-SwComponentTypes](#) "ThrottleActuator", "InjectionActuator", and "IgnitionActuator".

Engine Mode and Control The engine can be operated in different operation modes. The [AtomicSwComponentType](#) "OperatingMode" includes a state machine

which decides what setting for the application based calculation is used depending on the current mode, for example normal drive, idle speed, etc. Similar values are delivered by the "IdleSpeedControl" which determines important inputs for application calculation during idle speed.

Miscellaneous The [AtomicSwComponentType](#) "InjBatVoltCorrectionSensor" provides the input from the battery voltage sensor. The [AtomicSwComponentType](#) "CylNumObserver" is checking whether a change in the cylinder number is sensed and afterwards schedules the application based calculation. In this example application it is assumed that the cylinder number is provided externally within a rate of 2.5ms.

Since giving a complete overview of the system would result in an highly connected graph, Figure B.2 shows a simplified sketch of the [System](#) because a detailed presentation of such a system would go far beyond the scope of this section. The blue colored lines show important signal paths that are considered to be important for timing analysis and are typically subject to be constrained by timing requirements.

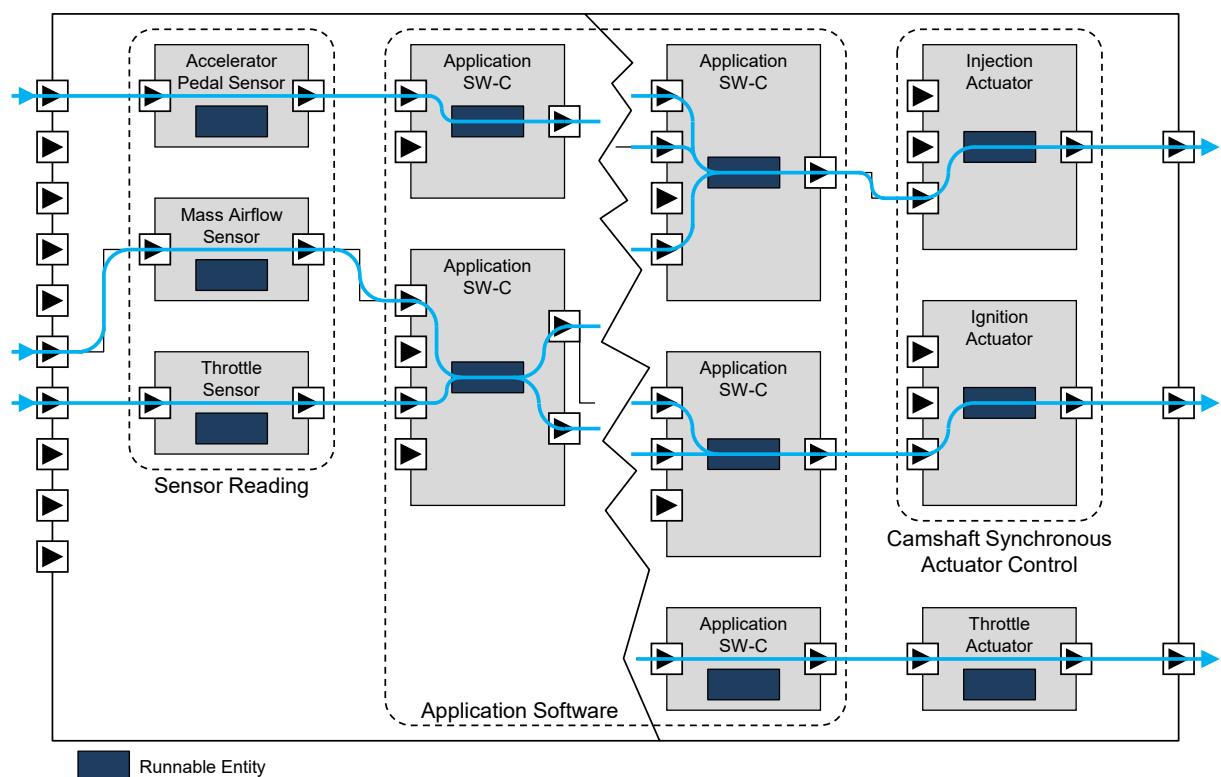


Figure B.2: Rough sketch of an internal combustion engine control system including important signal flow paths.

B.2.2 Timing Requirements

Assumed the following timing requirements are stated and imposed on the sketched engine control application:

1. When the position of the accelerator pedal changes then the throttle shall be actuated within 30ms.
2. The maximum age of the throttle position value tolerated by the application software shall not exceed 10ms.
3. The calculation of ignition timing shall be completed latest 50ms after the a change of the position of the accelerator pedal has been detected.
4. The calculation of the ignition timing shall be completed 3ms after the [BswIn-
terruptEntity](#) of the Basic Software Module called "Camshaft" has been activated.
5. For each cylinder the calculation of the corresponding injection mass shall be activated every 20ms (50Hz) and shall be completed not later than 20ms after its activation.

The listed requirements above need to be transformed into timing requirements captured by the capabilities of the AUTOSAR Specification of Timing Extensions. The following subsections present how the timing models look alike for every of those requirements.

B.2.3 Formal description of timing constraints in VFB View

It should be understood that the requirements from section [B.2.2](#) can be mapped to timing constraints that reference different parts of the system. Since a comprehensive and detailed overview of the whole system would go beyond the scope of this section only the important parts of the system and its timing are given to convey the idea behind using the AUTOSAR Specification of Timing Extensions for each presented timing requirement.

The requirements 1 to 3 are expressed in the VFB view respectively VFB Timing ([VfbTiming](#)).

B.2.3.1 Requirement 1

Figure [B.3](#) shows the simplified signal flow and involved components. It has been identified that the critical path of execution has an effect on four software components. The sensor software component "AcceleratorPedal" is responsible for reading in the signal and passes it to the application software component "AcceleratorPedalVoter". Afterwards the processed signal is further processed in the application software component "ThrottleController" until it is finally sent to the actuator via the actuator software component "ThrottleActuator". For specification of the timing constraint a timing description event chain shall be defined along with the appropriate timing description events. These timing descriptions and timing constraints are presented in listing [C.22](#) and cover the whole path from the sensor to the actuator.

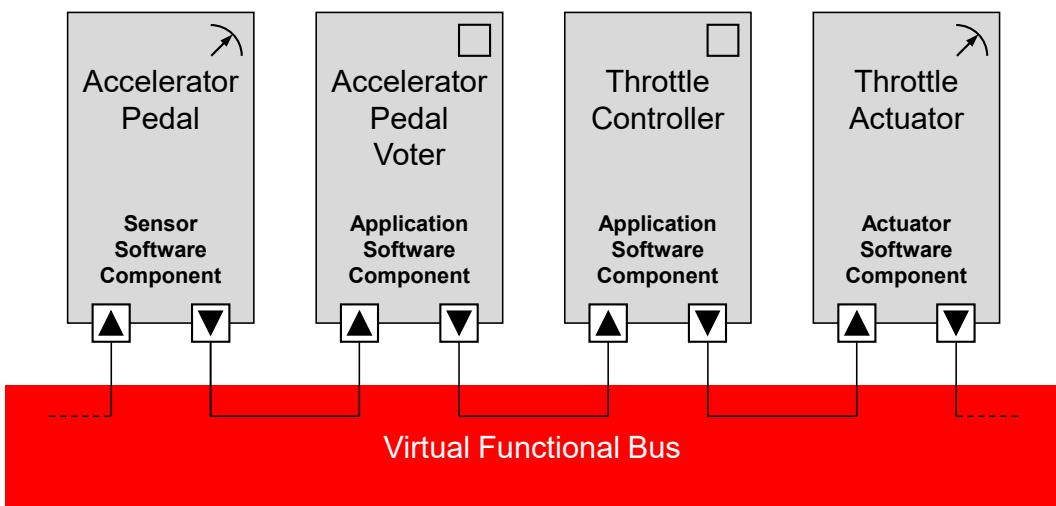


Figure B.3: Involved components for signal flow from "AcceleratorPedal" to "Throttle-Actuator" for timing requirement 1.

Since a timing constraint is imposed on the "sensor to actuator" communication the chosen constraint is a [LatencyTimingConstraint](#) and its type is "Reaction". Also note that the overall timing event chain references all event chain segments the event chain consists of.

B.2.3.2 Requirement 2

Requirement 2 specifies a typical timing constraint concerning the age of data provided by a sensor. For calculation in the [AtomicSwComponentType](#) called "BaseFuelMass" — which is here chosen as an example of the application software — a maximum age of input data concerning the throttle angle shall be guaranteed. The sensor value is determined in the [SensorActuatorSwComponentType](#) called "ThrottleSensor" and is passed to the application software. Figure B.4 shows all involved software components.

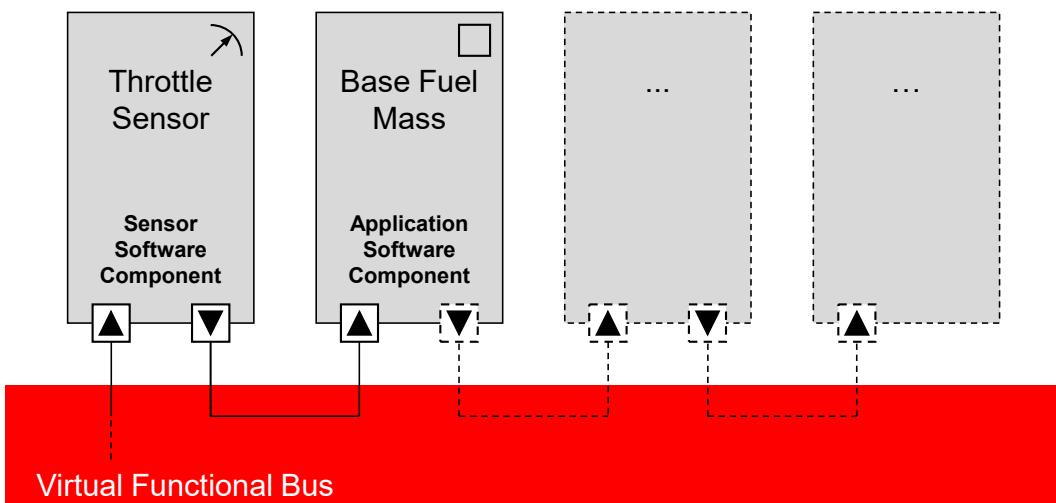


Figure B.4: Involved components for signal flow from "ThrottleSensor" to application component "BaseFuelMass" for timing requirement 2.

Please note that even if the signal flow continues to other parts of the system, it is possible to specify only this aspect of the desired timing behavior as shown in listing C.23.

B.2.3.3 Requirement 3

In requirement 3 a more complex timing description event chain is constrained. The first part of the event chain is already defined in the context of requirement 1. Thus, one can reference the set of defined events as well as the already specified timing description event chains. The second part of the event chain captures the feedback in the system that observes the sensor, in particular "ThrottleSensor". Please note that all events shall have a functional dependency, so it is important to understand that the [SensorActuatorSwComponentType](#) "ThrottleSensor" shall utilize up-to-date information of the [SensorActuatorSwComponentType](#) "ThrottleActuator". Figure B.5 shows the entire signal path and listing C.24 presents the entire timing information consisting of timing descriptions and timing constraints.

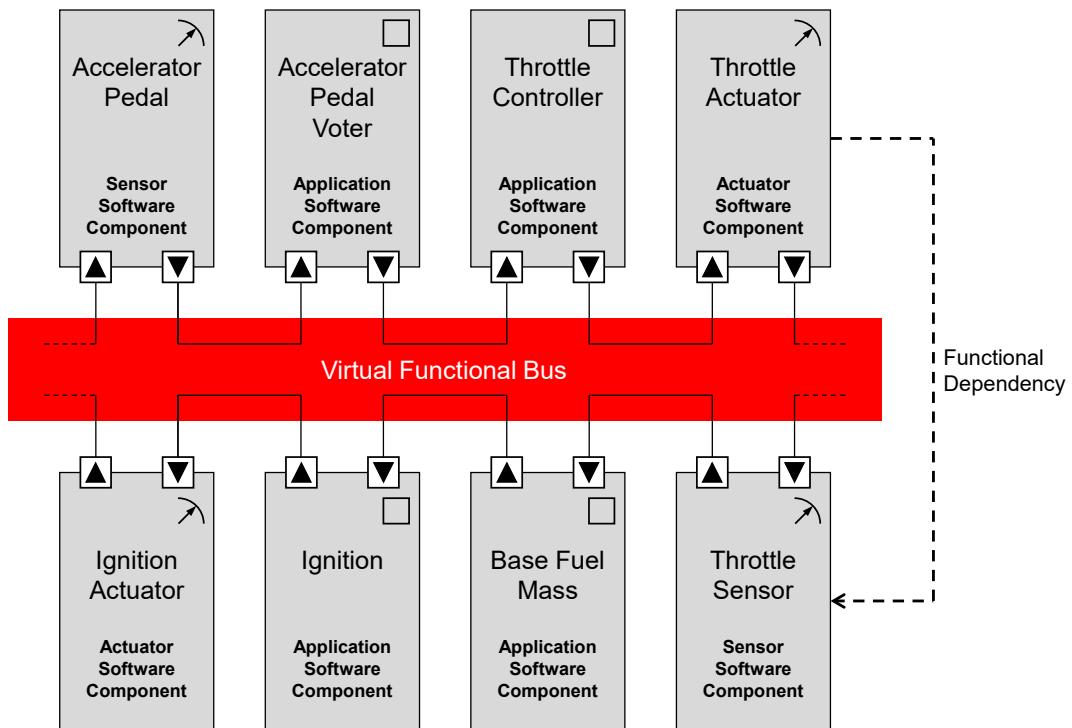


Figure B.5: Involved components for signal flow from sensor software component "AcceleratorPedal" to actuator software component "IgnitionActuator" for timing requirement 3.

B.2.4 Formal description of timing constraints in ECU View

Since requirement 4 references to events that are related to basic software modules, namely the interrupt system, the events shall be defined in the scope of the ECU View respectively ECU Timing ([EcuTiming](#)).

B.2.4.1 Requirement 4

The stimulus event of the timing description event chain for this requirements is the start of the [BswInterruptEntity](#) of the basic software module called "Camshaft". And as a result the runnable entity of the software component "IgnitionActuator" is activated. The response event of the timing description event chain is the termination of the [RunnableEntity](#) belonging to the software component "IgnitionActuator" as shown in Figure [B.6](#).

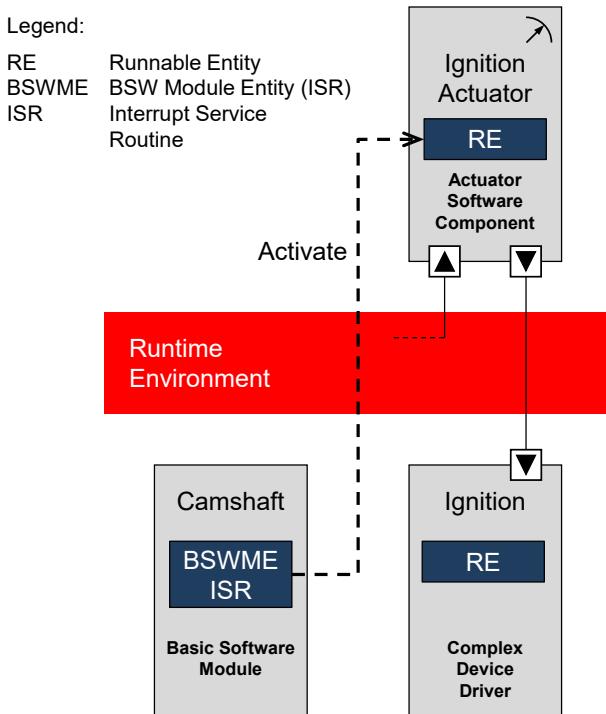


Figure B.6: Involved components and control flow for timing requirement 4.

The timing description events and event chains for this case are presented in listing C.25.

B.2.5 Formal description of timing constraints in SW-C View

Requirement 5 refers to execution behavior of a software component's [RunnableEntity](#) and therefore the scope is the Software Component (SW-C) View respectively Software Component (SW-C) Timing ([SwcTiming](#)).

B.2.5.1 Requirement 5

The SW-C timing references the internal behavior of [RunnableEntity](#) of SW-Cs. Here one reference the [RunnableEntity](#) of the software component "Ignition" which is a [ComplexDeviceDriverSwComponentType](#). In essence, the stated timing requirement requires firstly that the delay between activation and termination of the [RunnableEntity](#) is less than or equal 20 ms and secondly that the [RunnableEntity](#) is triggered at a frequency of 50 Hz which means that the runnable entity is periodically activated every 20 ms.

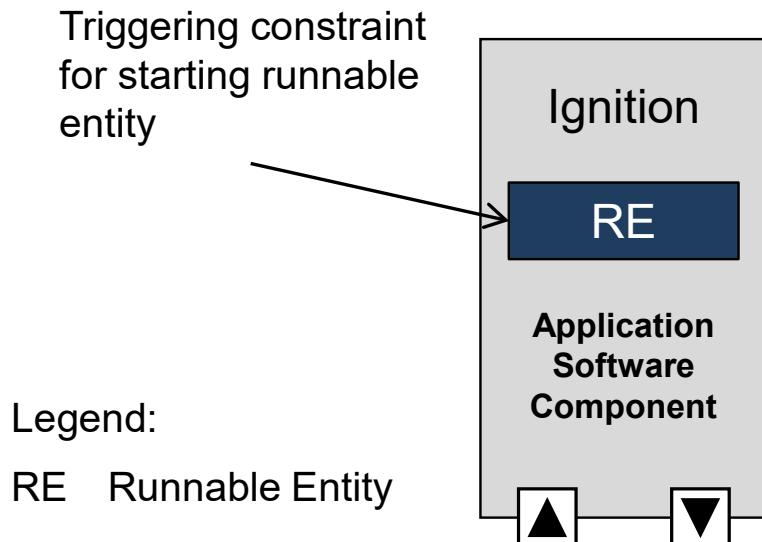


Figure B.7: Involved component and control flow for timing requirement 5.

B.3 Describing and Constraining Sensor and Actuator Timing

Chapter 2.3.2.1 describes the specification of VFB timing description events and introduces the attribute `isExternal` of such events. If the attribute is set to TRUE, then the event is considered to be *external*, which means that the event is supposed to occur on the physical sensor and/or actuator a `SensorActuatorSwComponentType`, a `ComplexDeviceDriverSwComponentType` and `EcuAbstractionSwComponentType` is dealing with. This chapter describes how this attribute is used to describe events for sensor and actuator timing, how the different events of such kind relate to each other in event chains, and how the timing can be constrained using `TimingConstraints`.

One of the important purposes of the Timing Extensions is to specify end-to-end timing constraints already in early development phases. However, in the VFB view there does not exist such elements like physical sensors, physical actuators, or other hardware related elements to attach events to. Therefore, timing description events related to the VFB View can be used to declare "external" events. For sensor and actuator timing four cases can be distinguished: external events can be observed between a `SensorActuatorSwComponentType` and a `ComplexDeviceDriverSwComponentType`, as well as between a `SensorActuatorSwComponentType` and an `EcuAbstractionSwComponentType`.

B.3.1 External Event of a Sensor accessed via S/R

In this case the `SensorActuatorSwComponentType` receives data from the `EcuAbstractionSwComponentType` or `ComplexDeviceDriverSwComponentType` through a sender-receiver interface via its required port. Two events `TDEventVariableDataPrototype`, indicating the receipt of data, are specified and both referencing the same required port and pointing to the the same `VariableDataPrototype`. The attribute `isExternal` of one of those events is set to TRUE and the same attribute of the other event is set to FALSE.

The semantics of the external event is that it occurs at the hardware level. The semantics of the other event is that it indicates the receipt of the data via the corresponding required port of the `SensorActuatorSwComponentType`. And the notion is that the external event occurs before the event indicating the receipt of data.

B.3.2 External Event of an Actuator accessed via S/R

In this case the `SensorActuatorSwComponentType` sends data to the `EcuAbstractionSwComponentType` or `ComplexDeviceDriverSwComponentType` through a S/R interface via its provided port. Two events `TDEventVariableDataPrototype`, indicating the sending of data, are specified and both referencing the same provided port and pointing to the the same `VariableDataPrototype`. The attribute `isExternal` of one of those events is set to TRUE and the same attribute of the other event is set to FALSE.

The semantics of the external event is that it occurs at the hardware level. The semantics of the other event is that it indicates the sending of the data via the corresponding provided port of the `SensorActuatorSwComponentType`. And the notion is that the event indicating the sending of data occurs before the external event.

B.3.3 External Event of a Sensor accessed via C/S

In this case the `SensorActuatorSwComponentType` receives data from the `EcuAbstractionSwComponentType` or `ComplexDeviceDriverSwComponentType` through a C/S interface on its required port. Two events `TDEventOperation`, indicating the receipt of the results of such an operation call, are specified and both referencing the same required port and pointing to the the same `ClientServerOperation`. The attribute `isExternal` of one of those events is set to TRUE and the same attribute of the other event is set to FALSE.

The semantics of the external event is that it occurs at the hardware level. The semantics of the other event is that it indicates the receipt of the data via the corresponding required port of the `SensorActuatorSwComponentType`. And the notion is that the external event occurs before the event indicating the receipt of the result of the operation call.

B.3.4 External Event of an Actuator accessed via C/S

In this case the `SensorActuatorSwComponentType` sends data to the `EcuAbstractionSwComponentType` or `ComplexDeviceDriverSwComponentType` through a C/S interface on its required port.

Two events `TDEventOperation`, indicating the invocation of the such an operation call, are specified and both referencing the same required port and pointing to the the same `ClientServerOperation`. The attribute `isExternal` of one of those events is set to TRUE and the same attribute of the other event is set to FALSE.

The semantics of the external event is that it occurs at the hardware level. The semantics of the other event is that it indicates the receipt of the data via the corresponding required port of the `SensorActuatorSwComponentType`. And the notion is that the event indicating the invocation of the operation occurs before the external event.

B.3.5 Considering hardware I/O latency of EventChains at VFB-level

To express an end-to-end sensor or actuator timing description event chain that also comprises hardware related latencies, already at VFB level, it is necessary to set the attribute `isExternal` of the stimulus and/or response accordingly. The overall end-to-end timing description event chain thus also comprises the "Input Latency" and/or the "Output Latency".

B.3.5.1 Input latency

The input latency is defined as the time latency between the point in time where the data is generated by a hardware I/O (e.g. a physical sensor) and the point in time where it is available for the application component, e.g. a `SensorActuatorSwComponentType`. The input latency is the time between the two events described in B.3.1 and B.3.3, respectively, depending on the communication type.

B.3.5.2 Output latency

The output latency is defined as the time latency between the point in time where the data is sent by the application component, e.g. a `SensorActuatorSwComponentType`, and the point in time where it is consumed by a hardware I/O (e.g. a physical actuator). The output latency is the time between the two events described in B.3.2 and B.3.4, respectively, depending on the communication type.

B.3.6 Constraining Input or Output Latency

The input or output latency can, for example, be modeled as event chain playing the role of a segment of the overall end-to-end chain. The overall end-to-end chain and also the input and output event chain segments can have attached timing constraints. This way either the overall end-to-end timing behavior or only the input and output behavior including hardware delay can be constrained already at VFB-level.

C Examples

C.1 Variant Handling

Sometimes it is necessary to specify that there are several alternatives with regard to timing requirements. For example, quite often it is reasonable to specify that a runnable entity shall be periodically activated either at 1ms, 2ms, 5ms, 8ms, or 10ms. In other words, it is perfectly fine to decide that the runnable entity is activated every 8ms. Indeed, it is allowed to activate the runnable entity either at: 1ms, 2ms, 5ms, 8ms, or 10ms. Hence, there should be a means to specify such time sets which contain all allowed timings, like in case of activating a runnable entity at $\{1, 2, 5, 8, 10\}$ ms.

For the purpose of specifying time sets the timing extensions utilize the Variant Handling capabilities specified and described in [8] chapter "Variant Handling".

As shown in Figure C.1 a SWC called `PlainVanillaSwc` consists of one runnable entity named `RunnableEntityOne`. This runnable entity calculates the value of the variable `DataElementTwo` based on the value of the variable `DataElementOne`.

The latter variable is received via the required port called `rPortOne` and the former one is written to the provided port named `pPortOne`. In the example three alternatives for activating the runnable entity are specified and shown in the table on the right hand side in Figure C.1.

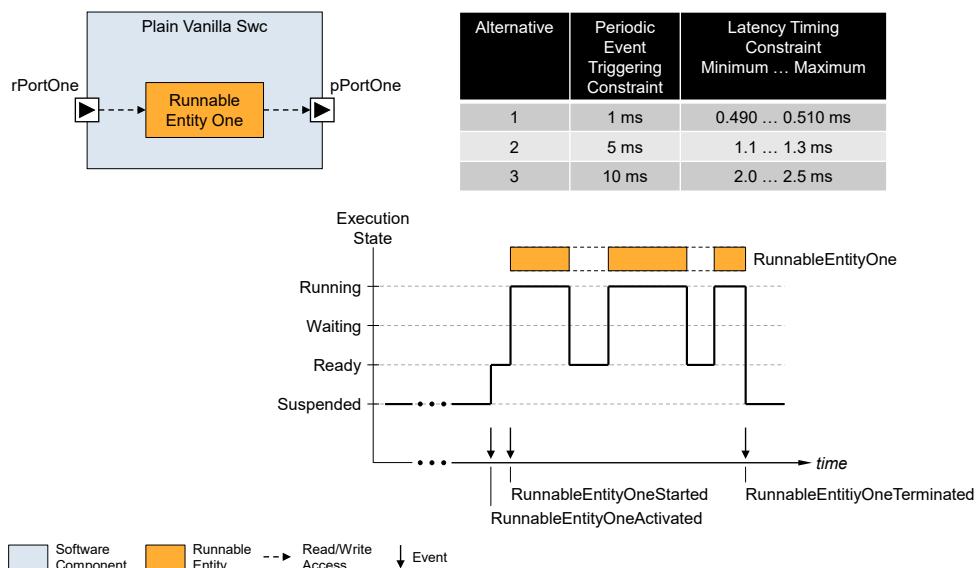


Figure C.1: Example of a set of timings specified for a Runnable Entity.

In listing C.1, the `SwcTiming` called `swctPlainVanillaSwc` specifies the timing descriptions and timing requirements. The timing descriptions describe the events that are related to the runnable entity of a SWC `PlainVanillaSwc`, in particular the activation event `RunnableEntityOneActivated`, start event `RunnableEntityOneStarted` and termination event `RunnableEntityOneTerminated` of the runnable

entity RunnableEntityOne. In addition, a timing description event chain CalculateValueOfDataElementTwo is specified that describes the causal relationship between the events RunnableEntityOneActivated and RunnableEntityOneTerminated.

The timing requirements specify three variants of the periodic event triggering constraint named PeriodicActivationRunnableEntityOne describing the *periodic* activation of the runnable entity RunnableEntityOne. The following alternatives are specified: 1ms, 5ms and 10ms. In addition, three alternatives for latency timing constraints, named ResponseTimeForCalculatingDataElementTwo, are specified that are imposed on the response time of the runnable entity. Technically, the scope of this timing constraint is the event chain CalculateValueOfDataElementTwo mentioned before.

The system constant called ScTimingRunnableEntityOne that eventually is used to select the specific variant of the required timing - the activation frequency and response time of the runnable entity. The selected variant is specified by the system constant's value in the [SwSystemconstantValueSet](#). In the given example the first variant Timing1 is selected.

```

<AR-PACKAGE>
  <SHORT-NAME>TimingExtensions</SHORT-NAME>
  <ELEMENTS>
    <SWC-TIMING>
      <SHORT-NAME>swctPlainVanillaSwc</SHORT-NAME>
      <TIMING-DESCRIPTIONS>
        <TD-EVENT-SWC-INTERNAL-BEHAVIOR>
          <SHORT-NAME>RunnableEntityOneActivated</SHORT-NAME>
          <RUNNABLE-REF DEST="RUNNABLE-ENTITY">/TimeSetExample/PlainVanillaSwc/InternalBehaviorPlainVanillaSwc/
            RunnableEntityOne</RUNNABLE-REF>
          <TD-EVENT-SWC-INTERNAL-BEHAVIOR-TYPE>RUNNABLE-ENTITY-ACTIVATED</TD-EVENT-SWC-INTERNAL-BEHAVIOR-TYPE>
        </TD-EVENT-SWC-INTERNAL-BEHAVIOR>
        <TD-EVENT-SWC-INTERNAL-BEHAVIOR>
          <SHORT-NAME>RunnableEntityOneTerminated</SHORT-NAME>
          <RUNNABLE-REF DEST="RUNNABLE-ENTITY">/TimeSetExample/PlainVanillaSwc/InternalBehaviorPlainVanillaSwc/
            RunnableEntityOne</RUNNABLE-REF>
          <TD-EVENT-SWC-INTERNAL-BEHAVIOR-TYPE>RUNNABLE-ENTITY-TERMINATED</TD-EVENT-SWC-INTERNAL-BEHAVIOR-TYPE>
        </TD-EVENT-SWC-INTERNAL-BEHAVIOR>
      <TIMING-DESCRIPTION-EVENT-CHAIN>
        <SHORT-NAME>CalculateValueOfDataElementTwo</SHORT-NAME>
        <STIMULUS-REF DEST="TD-EVENT-SWC-INTERNAL-BEHAVIOR">/TimingExtensions/swctPlainVanillaSwc/
          RunnableEntityOneActivated</STIMULUS-REF>
        <RESPONSE-REF DEST="TD-EVENT-SWC-INTERNAL-BEHAVIOR">/TimingExtensions/swctPlainVanillaSwc/
          RunnableEntityOneTerminated</RESPONSE-REF>
        <SEGMENT-REFS>
          <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/TimingExtensions/swctPlainVanillaSwc/
            CalculateValueOfDataElementTwo</SEGMENT-REF>
        </SEGMENT-REFS>
      </TIMING-DESCRIPTION-EVENT-CHAIN>
    </TIMING-DESCRIPTIONS>
    <TIMING-REQUIREMENTS>
      <PERIODIC-EVENT-TRIGGERING>
        <SHORT-NAME>PeriodicActivationRunnableEntityOne</SHORT-NAME>
        <VARIATION-POINT>
          <SHORT-LABEL>VpPetTiming1</SHORT-LABEL>
          <SW-SYSCOND BINDING-TIME="SYSTEM-DESIGN-TIME">(<SYSC-REF DEST="SW-SYSTEMCONST">/SystemConstants/
            ScTimingRunnableEntityOne</SYSC-REF>==1) </SW-SYSCOND>
        </VARIATION-POINT>
        <EVENT-REF DEST="TD-EVENT-SWC-INTERNAL-BEHAVIOR">/TimingExtensions/swctPlainVanillaSwc/
          RunnableEntityOneActivated</EVENT-REF>
      <PERIOD>
        <CSE-CODE>3</CSE-CODE>
        <CSE-CODE-FACTOR>1</CSE-CODE-FACTOR>
      </PERIOD>
      <PERIODIC-EVENT-TRIGGERING>
        <SHORT-NAME>PeriodicActivationRunnableEntityOne</SHORT-NAME>
        <VARIATION-POINT>
          <SHORT-LABEL>VpPetTiming2</SHORT-LABEL>
          <SW-SYSCOND BINDING-TIME="SYSTEM-DESIGN-TIME">(<SYSC-REF DEST="SW-SYSTEMCONST">/SystemConstants/
            ScTimingRunnableEntityOne</SYSC-REF>==2) </SW-SYSCOND>
        </VARIATION-POINT>
        <EVENT-REF DEST="TD-EVENT-SWC-INTERNAL-BEHAVIOR">/TimingExtensions/swctPlainVanillaSwc/
          RunnableEntityOneActivated</EVENT-REF>
      <PERIOD>
        <CSE-CODE>3</CSE-CODE>
        <CSE-CODE-FACTOR>5</CSE-CODE-FACTOR>
      </PERIOD>
    </TIMING-REQUIREMENTS>
  </ELEMENTS>
</AR-PACKAGE>

```

```

</PERIODIC-EVENT-TRIGGERING>
<PERIODIC-EVENT-TRIGGERING>
    <SHORT-NAME>PeriodicActivationRunnableEntityOne</SHORT-NAME>
    <VARIATION-POINT>
        <SHORT-LABEL>VpPetTiming3</SHORT-LABEL>
        <SW-SYSCOND BINDING-TIME="SYSTEM-DESIGN-TIME">(<SYSC-REF DEST="SW-SYSTEMCONST">/SystemConstants/
            ScTimingRunnableEntityOne</SYSC-REF>==3) </SW-SYSCOND>
    </VARIATION-POINT>
    <EVENT-REF DEST="TD-EVENT-SWC-INTERNAL-BEHAVIOR">/TimingExtensions/swctPlainVanillaSwc/
        RunnableEntityOneActivated</EVENT-REF>
    <PERIOD>
        <CSE-CODE>3</CSE-CODE>
        <CSE-CODE-FACTOR>10</CSE-CODE-FACTOR>
    </PERIOD>
</PERIODIC-EVENT-TRIGGERING>
<LATENCY-TIMING-CONSTRAINT>
    <SHORT-NAME>ResponseTimeForCalculatingDataElementTwo</SHORT-NAME>
    <VARIATION-POINT>
        <SHORT-LABEL>VpLtcTiming1</SHORT-LABEL>
        <SW-SYSCOND BINDING-TIME="SYSTEM-DESIGN-TIME">(<SYSC-REF DEST="SW-SYSTEMCONST">/SystemConstants/
            ScTimingRunnableEntityOne</SYSC-REF>==1) </SW-SYSCOND>
    </VARIATION-POINT>
    <LATENCY-CONSTRAINT-TYPE>REACTION</LATENCY-CONSTRAINT-TYPE>
    <SCOPE-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/TimingExtensions/swctPlainVanillaSwc/
        CalculateValueOfDataElementTwo</SCOPE-REF>
    <MINIMUM>
        <CSE-CODE>2</CSE-CODE>
        <CSE-CODE-FACTOR>490</CSE-CODE-FACTOR>
    </MINIMUM>
    <MAXIMUM>
        <CSE-CODE>2</CSE-CODE>
        <CSE-CODE-FACTOR>510</CSE-CODE-FACTOR>
    </MAXIMUM>
</LATENCY-TIMING-CONSTRAINT>
<LATENCY-TIMING-CONSTRAINT>
    <SHORT-NAME>ResponseTimeForCalculatingDataElementTwo</SHORT-NAME>
    <VARIATION-POINT>
        <SHORT-LABEL>VpLtcTiming2</SHORT-LABEL>
        <SW-SYSCOND BINDING-TIME="SYSTEM-DESIGN-TIME">(<SYSC-REF DEST="SW-SYSTEMCONST">/SystemConstants/
            ScTimingRunnableEntityOne</SYSC-REF>==2) </SW-SYSCOND>
    </VARIATION-POINT>
    <LATENCY-CONSTRAINT-TYPE>REACTION</LATENCY-CONSTRAINT-TYPE>
    <SCOPE-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/TimingExtensions/swctPlainVanillaSwc/
        CalculateValueOfDataElementTwo</SCOPE-REF>
    <MINIMUM>
        <CSE-CODE>2</CSE-CODE>
        <CSE-CODE-FACTOR>1100</CSE-CODE-FACTOR>
    </MINIMUM>
    <MAXIMUM>
        <CSE-CODE>2</CSE-CODE>
        <CSE-CODE-FACTOR>1300</CSE-CODE-FACTOR>
    </MAXIMUM>
</LATENCY-TIMING-CONSTRAINT>
<LATENCY-TIMING-CONSTRAINT>
    <SHORT-NAME>ResponseTimeForCalculatingDataElementTwo</SHORT-NAME>
    <VARIATION-POINT>
        <SHORT-LABEL>VpLtcTiming3</SHORT-LABEL>
        <SW-SYSCOND BINDING-TIME="SYSTEM-DESIGN-TIME">(<SYSC-REF DEST="SW-SYSTEMCONST">/SystemConstants/
            ScTimingRunnableEntityOne</SYSC-REF>==3) </SW-SYSCOND>
    </VARIATION-POINT>
    <LATENCY-CONSTRAINT-TYPE>REACTION</LATENCY-CONSTRAINT-TYPE>
    <SCOPE-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/TimingExtensions/swctPlainVanillaSwc/
        CalculateValueOfDataElementTwo</SCOPE-REF>
    <MINIMUM>
        <CSE-CODE>2</CSE-CODE>
        <CSE-CODE-FACTOR>2000</CSE-CODE-FACTOR>
    </MINIMUM>
    <MAXIMUM>
        <CSE-CODE>2</CSE-CODE>
        <CSE-CODE-FACTOR>2500</CSE-CODE-FACTOR>
    </MAXIMUM>
</LATENCY-TIMING-CONSTRAINT>
<TIMING-REQUIREMENTS>
    <BEHAVIOR-REF DEST="SWC-INTERNAL-BEHAVIOR">/TimeSetExample/PlainVanillaSwc/InternalBehaviorPlainVanillaSwc</BEHAVIOR
        -REF>
    </SWC-TIMING>
</ELEMENTS>
</AR-PACKAGE>
<AR-PACKAGE>
    <SHORT-NAME>SystemConstants</SHORT-NAME>
    <ELEMENTS>
        <SW-SYSTEMCONST>
            <SHORT-NAME>ScTimingRunnableEntityOne</SHORT-NAME>
        </SW-SYSTEMCONST>
    </ELEMENTS>
</AR-PACKAGE>
<AR-PACKAGE>
    <SHORT-NAME>Variants</SHORT-NAME>
    <ELEMENTS>
        <SW-SYSTEMCONSTANT-VALUE-SET>
            <SHORT-NAME>TIMEX_EXP_Time_Set_VariantValues</SHORT-NAME>
        <SW-SYSTEMCONSTANT-VALUES>
            <SW-SYSTEMCONST-VALUE>
                <!-- The values of the system constant ScTimingRunnableEntityOne may range from 1 to 3. -->
                <SW-SYSTEMCONST-REF DEST="SW-SYSTEMCONST">/SystemConstants/ScTimingRunnableEntityOne</SW-SYSTEMCONST-REF>
            </SW-SYSTEMCONST-VALUE>
        </SW-SYSTEMCONSTANT-VALUES>
    </ELEMENTS>
</AR-PACKAGE>

```

```

<VALUE>1</VALUE>
</SW-SYSTEMCONST-VALUE>
</SW-SYSTEMCONSTANT-VALUES>
</SW-SYSTEMCONSTANT-VALUE-SET>
<POST-BUILD-VARIANT-CRITERION-VALUE-SET>
  <SHORT-NAME>TIMEX_EXP_Time_Set_VariantPostBuildValues</SHORT-NAME>
</POST-BUILD-VARIANT-CRITERION-VALUE-SET>
<PREDEFINED-VARIANT>
  <SHORT-NAME>TIMEX_EXP_Time_Set_Variant</SHORT-NAME>
  <POST-BUILD-VARIANT-CRITERION-VALUE-SET-REFS>
    <POST-BUILD-VARIANT-CRITERION-VALUE-SET-REF DEST="POST-BUILD-VARIANT-CRITERION-VALUE-SET">/Variants/
      TIMEX_EXP_Time_Set_VariantPostBuildValues</POST-BUILD-VARIANT-CRITERION-VALUE-SET-REF>
    </POST-BUILD-VARIANT-CRITERION-VALUE-SET-REFS>
  <SW-SYSTEMCONSTANT-VALUE-SET-REFS>
    <SW-SYSTEMCONSTANT-VALUE-SET-REF DEST="SW-SYSTEMCONSTANT-VALUE-SET">/Variants/TIMEX_EXP_Time_Set_VariantValues</SW
      -SYSTEMCONSTANT-VALUE-SET-REF>
    </SW-SYSTEMCONSTANT-VALUE-SET-REFS>
  </PREDEFINED-VARIANT>
</ELEMENTS>
</AR-PACKAGE>
  
```

Listing C.1: ARXML for Variant Handling

C.2 Logical Execution Time

C.2.1 Logical Execution Time - Timing descriptions

```

<TD-EVENT-COMPLEX>
  <SHORT-NAME>LetIntervalRelease</SHORT-NAME>
  <CATEGORY>LET_RELEASE</CATEGORY>
</TD-EVENT-COMPLEX>
<TD-EVENT-COMPLEX>
  <SHORT-NAME>LetIntervalTerminate</SHORT-NAME>
  <CATEGORY>LET_TERMINATE</CATEGORY>
</TD-EVENT-COMPLEX>
<TIMING-DESCRIPTION-EVENT-CHAIN>
  <SHORT-NAME>tdecLetInterval</SHORT-NAME>
  <CATEGORY>LET_INTERVAL</CATEGORY>
  <STIMULUS-REF DEST="TD-EVENT-COMPLEX">/LogicalExecutionTime/TimingExtensions/LogicalExecutionTime02/
    LetIntervalRelease</STIMULUS-REF>
  <RESPONSE-REF DEST="TD-EVENT-COMPLEX">/LogicalExecutionTime/TimingExtensions/LogicalExecutionTime02/
    LetIntervalTerminate</RESPONSE-REF>
  <SEGMENT-REFS>
    <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/LogicalExecutionTime/TimingExtensions/
      LogicalExecutionTime02/tdecLetInterval</SEGMENT-REF>
  </SEGMENT-REFS>
</TIMING-DESCRIPTION-EVENT-CHAIN>
  
```

Listing C.2: Specifying LET Interval

C.2.2 Logical Execution Time - Timing constraints - Latency Timing

```

<LATENCY-TIMING-CONSTRAINT>
  <SHORT-NAME>ltcLetInterval</SHORT-NAME>
  <CATEGORY>LET_INTERVAL</CATEGORY>
  <LATENCY-CONSTRAINT-TYPE>REACTION</LATENCY-CONSTRAINT-TYPE>
  <SCOPE-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/LogicalExecutionTime/TimingExtensions/
    LogicalExecutionTime02/tdecLetInterval</SCOPE-REF>
  <MINIMUM>
    <CSE-CODE>3</CSE-CODE>
    <CSE-CODE-FACTOR>10</CSE-CODE-FACTOR>
  </MINIMUM>
  <MAXIMUM>
    <CSE-CODE>3</CSE-CODE>
    <CSE-CODE-FACTOR>10</CSE-CODE-FACTOR>
  </MAXIMUM>
</LATENCY-TIMING-CONSTRAINT>
  
```

Listing C.3: Specifying LET Interval Timing

C.2.3 Logical Execution Time - Timing constraints - Event Triggering

```

<PERIODIC-EVENT-TRIGGERING>
  <SHORT-NAME>petcLetIntervalRecurrence</SHORT-NAME>
  <EVENT-REF DEST="TD-EVENT-COMPLEX">/LogicalExecutionTime/TimingExtensions/LogicalExecutionTime02/
    LetIntervalRelease</EVENT-REF>
  <MINIMUM-INTER-ARRIVAL-TIME>
  
```

```

<CSE-CODE>3</CSE-CODE>
<CSE-CODE-FACTOR>10</CSE-CODE-FACTOR>
</MINIMUM-INTER-ARRIVAL-TIME>
<JITTER>
  <CSE-CODE>3</CSE-CODE>
  <CSE-CODE-FACTOR>0</CSE-CODE-FACTOR>
</JITTER>
<PERIOD>
  <CSE-CODE>3</CSE-CODE>
  <CSE-CODE-FACTOR>10</CSE-CODE-FACTOR>
</PERIOD>
</PERIODIC-EVENT-TRIGGERING>
  
```

Listing C.4: Specifying Recurrence of a LET Interval

C.2.4 Logical Execution Time - Timing constraints - Event Triggering - Gap

```

<LATENCY-TIMING-CONSTRAINT>
  <SHORT-NAME>ltcLetInterval_1</SHORT-NAME>
  <CATEGORY>LET_INTERVAL</CATEGORY>
  <LATENCY-CONSTRAINT-TYPE>REACTION</LATENCY-CONSTRAINT-TYPE>
  <SCOPE-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/SpecifyingPeriodicLetIntervalWithGap/
    sysTiming_SpecifyingPeriodicLetIntervalWithGap/tdecLetInterval_1</SCOPE-REF>
  <MINIMUM>
    <CSE-CODE>3</CSE-CODE>
    <CSE-CODE-FACTOR>6</CSE-CODE-FACTOR>
  </MINIMUM>
  <MAXIMUM>
    <CSE-CODE>3</CSE-CODE>
    <CSE-CODE-FACTOR>6</CSE-CODE-FACTOR>
  </MAXIMUM>
</LATENCY-TIMING-CONSTRAINT>
<PERIODIC-EVENT-TRIGGERING>
  <SHORT-NAME>petcLetInterval_1</SHORT-NAME>
  <MINIMUM-INTER-ARRIVAL-TIME>
    <CSE-CODE>3</CSE-CODE>
    <CSE-CODE-FACTOR>10</CSE-CODE-FACTOR>
  </MINIMUM-INTER-ARRIVAL-TIME>
  <JITTER>
    <CSE-CODE>3</CSE-CODE>
    <CSE-CODE-FACTOR>0</CSE-CODE-FACTOR>
  </JITTER>
  <PERIOD>
    <CSE-CODE>3</CSE-CODE>
    <CSE-CODE-FACTOR>10</CSE-CODE-FACTOR>
  </PERIOD>
</PERIODIC-EVENT-TRIGGERING>
  
```

Listing C.5: Periodic LET Interval with Gap

C.2.5 Logical Execution Time - Timing constraints - Offset Timing - Without

```

<OFFSET-TIMING-CONSTRAINT>
  <SHORT-NAME>otc_LetInterval_2</SHORT-NAME>
  <SOURCE-REF DEST="TD-EVENT-COMPLEX">/SpecifyingPeriodicLetIntervalsWithNoOffset/
    sysTiming_SpecifyingPeriodicLetIntervalsWithNoOffset/LetIntervalRelease_1</SOURCE-REF>
  <TARGET-REF DEST="TD-EVENT-COMPLEX">/SpecifyingPeriodicLetIntervalsWithNoOffset/
    sysTiming_SpecifyingPeriodicLetIntervalsWithNoOffset/LetIntervalRelease_2</TARGET-REF>
  <MINIMUM>
    <CSE-CODE>2</CSE-CODE>
    <CSE-CODE-FACTOR>0</CSE-CODE-FACTOR>
  </MINIMUM>
  <MAXIMUM>
    <CSE-CODE>2</CSE-CODE>
    <CSE-CODE-FACTOR>0</CSE-CODE-FACTOR>
  </MAXIMUM>
</OFFSET-TIMING-CONSTRAINT>
<OFFSET-TIMING-CONSTRAINT>
  <SHORT-NAME>otc_LetInterval_3</SHORT-NAME>
  <SOURCE-REF DEST="TD-EVENT-COMPLEX">/SpecifyingPeriodicLetIntervalsWithNoOffset/
    sysTiming_SpecifyingPeriodicLetIntervalsWithNoOffset/LetIntervalRelease_1</SOURCE-REF>
  <TARGET-REF DEST="TD-EVENT-COMPLEX">/SpecifyingPeriodicLetIntervalsWithNoOffset/
    sysTiming_SpecifyingPeriodicLetIntervalsWithNoOffset/LetIntervalRelease_3</TARGET-REF>
  <MINIMUM>
    <CSE-CODE>2</CSE-CODE>
    <CSE-CODE-FACTOR>0</CSE-CODE-FACTOR>
  </MINIMUM>
  <MAXIMUM>
    <CSE-CODE>2</CSE-CODE>
    <CSE-CODE-FACTOR>0</CSE-CODE-FACTOR>
  </MAXIMUM>
</OFFSET-TIMING-CONSTRAINT>
  
```

Listing C.6: Periodic LET Intervals with no Offset

C.2.6 Logical Execution Time - Timing constraints - Offset Timing - With

```

<OFFSET-TIMING-CONSTRAINT>
  <SHORT-NAME>otc_LetInterval_2</SHORT-NAME>
  <SOURCE-REF DEST="TD-EVENT-COMPLEX">/SpecifyingPeriodicLetIntervalsWithOffset/
    sysTiming_SpecifyingPeriodicLetIntervalsWithOffset/LetIntervalRelease_1</SOURCE-REF>
  <TARGET-REF DEST="TD-EVENT-COMPLEX">/SpecifyingPeriodicLetIntervalsWithOffset/
    sysTiming_SpecifyingPeriodicLetIntervalsWithOffset/LetIntervalRelease_2</TARGET-REF>
  <MINIMUM>
    <CSE-CODE>2</CSE-CODE>
    <CSE-CODE-FACTOR>25</CSE-CODE-FACTOR>
  </MINIMUM>
  <MAXIMUM>
    <CSE-CODE>2</CSE-CODE>
    <CSE-CODE-FACTOR>25</CSE-CODE-FACTOR>
  </MAXIMUM>
</OFFSET-TIMING-CONSTRAINT>
<OFFSET-TIMING-CONSTRAINT>
  <SHORT-NAME>otc_LetInterval_3</SHORT-NAME>
  <SOURCE-REF DEST="TD-EVENT-COMPLEX">/SpecifyingPeriodicLetIntervalsWithOffset/
    sysTiming_SpecifyingPeriodicLetIntervalsWithOffset/LetIntervalRelease_1</SOURCE-REF>
  <TARGET-REF DEST="TD-EVENT-COMPLEX">/SpecifyingPeriodicLetIntervalsWithOffset/
    sysTiming_SpecifyingPeriodicLetIntervalsWithOffset/LetIntervalRelease_3</TARGET-REF>
  <MINIMUM>
    <CSE-CODE>2</CSE-CODE>
    <CSE-CODE-FACTOR>75</CSE-CODE-FACTOR>
  </MINIMUM>
  <MAXIMUM>
    <CSE-CODE>2</CSE-CODE>
    <CSE-CODE-FACTOR>75</CSE-CODE-FACTOR>
  </MAXIMUM>
</OFFSET-TIMING-CONSTRAINT>

```

Listing C.7: Periodic LET Intervals with Offset

C.2.7 Logical Execution Time - Timing constraints - Offset Timing - Arbitrary

```

<OFFSET-TIMING-CONSTRAINT>
  <SHORT-NAME>otc_LetInterval_2</SHORT-NAME>
  <SOURCE-REF DEST="TD-EVENT-COMPLEX">/SpecifyingArbitraryLetIntervals/sysTiming_SpecifyingArbitraryLetIntervals/
    LetIntervalRelease_1</SOURCE-REF>
  <TARGET-REF DEST="TD-EVENT-COMPLEX">/SpecifyingArbitraryLetIntervals/sysTiming_SpecifyingArbitraryLetIntervals/
    LetIntervalRelease_2</TARGET-REF>
  <MINIMUM>
    <CSE-CODE>3</CSE-CODE>
    <CSE-CODE-FACTOR>2</CSE-CODE-FACTOR>
  </MINIMUM>
  <MAXIMUM>
    <CSE-CODE>3</CSE-CODE>
    <CSE-CODE-FACTOR>2</CSE-CODE-FACTOR>
  </MAXIMUM>
</OFFSET-TIMING-CONSTRAINT>
<OFFSET-TIMING-CONSTRAINT>
  <SHORT-NAME>otc_LetInterval_3</SHORT-NAME>
  <SOURCE-REF DEST="TD-EVENT-COMPLEX">/SpecifyingArbitraryLetIntervals/sysTiming_SpecifyingArbitraryLetIntervals/
    LetIntervalRelease_1</SOURCE-REF>
  <TARGET-REF DEST="TD-EVENT-COMPLEX">/SpecifyingArbitraryLetIntervals/sysTiming_SpecifyingArbitraryLetIntervals/
    LetIntervalRelease_3</TARGET-REF>
  <MINIMUM>
    <CSE-CODE>3</CSE-CODE>
    <CSE-CODE-FACTOR>8</CSE-CODE-FACTOR>
  </MINIMUM>
  <MAXIMUM>
    <CSE-CODE>3</CSE-CODE>
    <CSE-CODE-FACTOR>8</CSE-CODE-FACTOR>
  </MAXIMUM>
</OFFSET-TIMING-CONSTRAINT>
<OFFSET-TIMING-CONSTRAINT>
  <SHORT-NAME>otc_LetInterval_4</SHORT-NAME>
  <SOURCE-REF DEST="TD-EVENT-COMPLEX">/SpecifyingArbitraryLetIntervals/sysTiming_SpecifyingArbitraryLetIntervals/
    LetIntervalRelease_1</SOURCE-REF>
  <TARGET-REF DEST="TD-EVENT-COMPLEX">/SpecifyingArbitraryLetIntervals/sysTiming_SpecifyingArbitraryLetIntervals/
    LetIntervalRelease_4</TARGET-REF>
  <MINIMUM>
    <CSE-CODE>3</CSE-CODE>
    <CSE-CODE-FACTOR>9</CSE-CODE-FACTOR>
  </MINIMUM>
  <MAXIMUM>
    <CSE-CODE>3</CSE-CODE>
    <CSE-CODE-FACTOR>9</CSE-CODE-FACTOR>
  </MAXIMUM>
</OFFSET-TIMING-CONSTRAINT>

```

Listing C.8: Arbitrary LET Intervals

C.2.8 Logical Execution Time - Timing constraints - Offset Timing - Arbitrary Overlap

```

<OFFSET-TIMING-CONSTRAINT>
  <SHORT-NAME>otc_LetInterval_2</SHORT-NAME>
  <SOURCE-REF DEST="TD-EVENT-COMPLEX">/SpecifyingArbitraryLetIntervalsWithOverlap/
    sysTiming_SpecifyingArbitraryLetIntervalsWithOverlap/LetIntervalRelease_1</SOURCE-REF>
  <TARGET-REF DEST="TD-EVENT-COMPLEX">/SpecifyingArbitraryLetIntervalsWithOverlap/
    sysTiming_SpecifyingArbitraryLetIntervalsWithOverlap/LetIntervalRelease_2</TARGET-REF>
  <MINIMUM>
    <CSE-CODE>3</CSE-CODE>
    <CSE-CODE-FACTOR>13</CSE-CODE-FACTOR>
  </MINIMUM>
  <MAXIMUM>
    <CSE-CODE>3</CSE-CODE>
    <CSE-CODE-FACTOR>13</CSE-CODE-FACTOR>
  </MAXIMUM>
</OFFSET-TIMING-CONSTRAINT>
<OFFSET-TIMING-CONSTRAINT>
  <SHORT-NAME>otc_LetInterval_3</SHORT-NAME>
  <SOURCE-REF DEST="TD-EVENT-COMPLEX">/SpecifyingArbitraryLetIntervalsWithOverlap/
    sysTiming_SpecifyingArbitraryLetIntervalsWithOverlap/LetIntervalRelease_1</SOURCE-REF>
  <TARGET-REF DEST="TD-EVENT-COMPLEX">/SpecifyingArbitraryLetIntervalsWithOverlap/
    sysTiming_SpecifyingArbitraryLetIntervalsWithOverlap/LetIntervalRelease_3</TARGET-REF>
  <MINIMUM>
    <CSE-CODE>3</CSE-CODE>
    <CSE-CODE-FACTOR>4</CSE-CODE-FACTOR>
  </MINIMUM>
  <MAXIMUM>
    <CSE-CODE>3</CSE-CODE>
    <CSE-CODE-FACTOR>4</CSE-CODE-FACTOR>
  </MAXIMUM>
</OFFSET-TIMING-CONSTRAINT>
<OFFSET-TIMING-CONSTRAINT>
  <SHORT-NAME>otc_LetInterval_4</SHORT-NAME>
  <SOURCE-REF DEST="TD-EVENT-COMPLEX">/SpecifyingArbitraryLetIntervalsWithOverlap/
    sysTiming_SpecifyingArbitraryLetIntervalsWithOverlap/LetIntervalRelease_1</SOURCE-REF>
  <TARGET-REF DEST="TD-EVENT-COMPLEX">/SpecifyingArbitraryLetIntervalsWithOverlap/
    sysTiming_SpecifyingArbitraryLetIntervalsWithOverlap/LetIntervalRelease_4</TARGET-REF>
  <MINIMUM>
    <CSE-CODE>3</CSE-CODE>
    <CSE-CODE-FACTOR>3</CSE-CODE-FACTOR>
  </MINIMUM>
  <MAXIMUM>
    <CSE-CODE>3</CSE-CODE>
    <CSE-CODE-FACTOR>3</CSE-CODE-FACTOR>
  </MAXIMUM>
</OFFSET-TIMING-CONSTRAINT>

```

Listing C.9: Arbitrary LET Intervals with Overlap

C.2.9 Logical Execution Time - Timing constraints - Execution Order

```

<EOC-EXECUTABLE-ENTITY-REF-GROUP>
  <SHORT-NAME>eocEerg_ExecutableEntitiesCluster_01</SHORT-NAME>
  <LET-INTERVAL-REFS>
    <LET-INTERVAL-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/MappingEecToLetInterval/
      sysTiming_MappingEecToLetInterval/tdecLetInterval_1</LET-INTERVAL-REF>
  </LET-INTERVAL-REFS>
  <NESTED-ELEMENT-REFS>
    <NESTED-ELEMENT-REF DEST="EOC-EVENT-REF">/MappingEecToLetInterval/sysTiming_MappingEecToLetInterval/
      eoc_DefiningAndMappingEecs/eocERef_te_re01_Swc01_1ms</NESTED-ELEMENT-REF>
    <NESTED-ELEMENT-REF DEST="EOC-EVENT-REF">/MappingEecToLetInterval/sysTiming_MappingEecToLetInterval/
      eoc_DefiningAndMappingEecs/eocERef_te_re01_Swc02_1ms</NESTED-ELEMENT-REF>
    <NESTED-ELEMENT-REF DEST="EOC-EVENT-REF">/MappingEecToLetInterval/sysTiming_MappingEecToLetInterval/
      eoc_DefiningAndMappingEecs/eocERef_te_re01_Swc04_1ms</NESTED-ELEMENT-REF>
  </NESTED-ELEMENT-REFS>
  <SUCCESSOR-REFS>
    <SUCCESSOR-REF DEST="EOC-EXECUTABLE-ENTITY-REF-GROUP">/MappingEecToLetInterval/
      sysTiming_MappingEecToLetInterval/eoc_DefiningAndMappingEecs/eocEerg_ExecutableEntitiesCluster_02</
      SUCCESSOR-REF>
    <SUCCESSOR-REF DEST="EOC-EXECUTABLE-ENTITY-REF-GROUP">/MappingEecToLetInterval/
      sysTiming_MappingEecToLetInterval/eoc_DefiningAndMappingEecs/eocEerg_ExecutableEntitiesCluster_03</
      SUCCESSOR-REF>
  </SUCCESSOR-REFS>

```

Listing C.10: Mapping Executable Entities Cluster to LET Intervals

C.3 Software Cluster

C.3.1 Software Cluster - Timing descriptions

```

<?xml version="1.0" encoding="UTF-8"?>
<AUTOSAR xmlns="http://autosar.org/schema/r4.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://autosar.org/schema/r4.0/AUTOSAR_00052.xsd">
<AR-PACKAGES>
    <AR-PACKAGE UID="bdbade53-5890-4e1c-aac0-91d3e71b2e19">
        <SHORT-NAME>System</SHORT-NAME>
        <ELEMENTS>
            <SYSTEM S="">
                <SHORT-NAME>SoftwareClusterA</SHORT-NAME>
                <CATEGORY>SW_CLUSTER_SYSTEM_DESCRIPTION</CATEGORY>
                <SW-CLUSTERS>
                    <CP-SOFTWARE-CLUSTER-REF-CONDITIONAL>
                        <CP-SOFTWARE-CLUSTER-REF DEST="CP-SOFTWARE-CLUSTER">/SoftwareClusters/SoftwareClusterA</CP-SOFTWARE-CLUSTER-REF>
                    </CP-SOFTWARE-CLUSTER-REF-CONDITIONAL>
                </SW-CLUSTERS>
            </SYSTEM>
            <SYSTEM S="">
                <SHORT-NAME>SoftwareClusterB</SHORT-NAME>
                <CATEGORY>SW_CLUSTER_SYSTEM_DESCRIPTION</CATEGORY>
                <SW-CLUSTERS>
                    <CP-SOFTWARE-CLUSTER-REF-CONDITIONAL>
                        <CP-SOFTWARE-CLUSTER-REF DEST="CP-SOFTWARE-CLUSTER">/SoftwareClusters/SoftwareClusterB</CP-SOFTWARE-CLUSTER-REF>
                    </CP-SOFTWARE-CLUSTER-REF-CONDITIONAL>
                </SW-CLUSTERS>
            </SYSTEM>
            <SYSTEM S="">
                <SHORT-NAME>SoftwareClusterHost</SHORT-NAME>
                <CATEGORY>SW_CLUSTER_SYSTEM_DESCRIPTION</CATEGORY>
                <SW-CLUSTERS>
                    <CP-SOFTWARE-CLUSTER-REF-CONDITIONAL>
                        <CP-SOFTWARE-CLUSTER-REF DEST="CP-SOFTWARE-CLUSTER">/SoftwareClusters/SoftwareClusterHost</CP-SOFTWARE-CLUSTER-REF>
                    </CP-SOFTWARE-CLUSTER-REF-CONDITIONAL>
                </SW-CLUSTERS>
            </SYSTEM>
            <SYSTEM>
                <SHORT-NAME>System</SHORT-NAME>
                <CATEGORY>SYSTEM_DESCRIPTION</CATEGORY>
                <ROOT-SOFTWARE-COMPOSITIONS>
                    <ROOT-SW-COMPOSITION-PROTOTYPE>
                        <SHORT-NAME>rscp_System</SHORT-NAME>
                        <SOFTWARE-COMPOSITION-TREF DEST="COMPOSITION-SW-COMPONENT-TYPE">/SwComponents/CSWCT01</SOFTWARE-COMPOSITION-TREF>
                    </ROOT-SW-COMPOSITION-PROTOTYPE>
                </ROOT-SOFTWARE-COMPOSITIONS>
            </SYSTEM>
        <ELEMENTS>
    </AR-PACKAGE>
    <AR-PACKAGE>
        <SHORT-NAME>SoftwareClusters</SHORT-NAME>
        <ELEMENTS>
            <CP-SOFTWARE-CLUSTER>
                <SHORT-NAME>SoftwareClusterA</SHORT-NAME>
                <CATEGORY>APPLICATION_SOFTWARE_CLUSTER</CATEGORY>
                <SW-COMPOSITIONS>
                    <COMPOSITION-SW-COMPONENT-TYPE-REF-CONDITIONAL>
                        <COMPOSITION-SW-COMPONENT-TYPE-REF DEST="COMPOSITION-SW-COMPONENT-TYPE">/SwComponents/CSWCT04</COMPOSITION-SW-COMPONENT-TYPE-REF>
                    </COMPOSITION-SW-COMPONENT-TYPE-REF-CONDITIONAL>
                    <COMPOSITION-SW-COMPONENT-TYPE-REF-CONDITIONAL>
                        <COMPOSITION-SW-COMPONENT-TYPE-REF DEST="COMPOSITION-SW-COMPONENT-TYPE">/SwComponents/CSWCT05</COMPOSITION-SW-COMPONENT-TYPE-REF>
                    </COMPOSITION-SW-COMPONENT-TYPE-REF-CONDITIONAL>
                    <COMPOSITION-SW-COMPONENT-TYPE-REF-CONDITIONAL>
                        <COMPOSITION-SW-COMPONENT-TYPE-REF DEST="COMPOSITION-SW-COMPONENT-TYPE">/SwComponents/CSWCT10</COMPOSITION-SW-COMPONENT-TYPE-REF>
                    </COMPOSITION-SW-COMPONENT-TYPE-REF-CONDITIONAL>
                </SW-COMPOSITIONS>
            </CP-SOFTWARE-CLUSTER>
            <CP-SOFTWARE-CLUSTER>
                <SHORT-NAME>SoftwareClusterB</SHORT-NAME>
                <CATEGORY>APPLICATION_SOFTWARE_CLUSTER</CATEGORY>
                <SW-COMPOSITIONS>
                    <COMPOSITION-SW-COMPONENT-TYPE-REF-CONDITIONAL>
                        <COMPOSITION-SW-COMPONENT-TYPE-REF DEST="COMPOSITION-SW-COMPONENT-TYPE">/SwComponents/CSWCT09</COMPOSITION-SW-COMPONENT-TYPE-REF>
                    </COMPOSITION-SW-COMPONENT-TYPE-REF-CONDITIONAL>
                    <COMPOSITION-SW-COMPONENT-TYPE-REF-CONDITIONAL>
                        <COMPOSITION-SW-COMPONENT-TYPE-REF DEST="COMPOSITION-SW-COMPONENT-TYPE">/SwComponents/CSWCT11</COMPOSITION-SW-COMPONENT-TYPE-REF>
                    </COMPOSITION-SW-COMPONENT-TYPE-REF-CONDITIONAL>
                </SW-COMPOSITIONS>
            </CP-SOFTWARE-CLUSTER>
            <CP-SOFTWARE-CLUSTER>
                <SHORT-NAME>SoftwareClusterHost</SHORT-NAME>
                <CATEGORY>HOST_SOFTWARE_CLUSTER</CATEGORY>
                <SW-COMPOSITIONS/>
            </CP-SOFTWARE-CLUSTER>
        <ELEMENTS>
    </AR-PACKAGE>
    <AR-PACKAGE>
        <SHORT-NAME>SoftwareClusterMappings</SHORT-NAME>
        <AR-PACKAGES>
            <AR-PACKAGE>
                <SHORT-NAME>AtoHost</SHORT-NAME>

```

```

<ELEMENTS>
  <TD-CP-SOFTWARE-CLUSTER-MAPPING-SET>
    <SHORT-NAME>AToHostMappingSet</SHORT-NAME>
    <TD-CP-SOFTWARE-CLUSTER-TO-TD-MAPPINGS>
      <TD-CP-SOFTWARE-CLUSTER-MAPPING>
        <SHORT-NAME>AToDispatcher10msMapping</SHORT-NAME>
        <PROVIDER-REF DEST="CP-SOFTWARE-CLUSTER">/SoftwareClusters/SoftwareClusterHost</PROVIDER-REF>
        <REQUESTOR-REFS>
          <REQUESTOR-REF DEST="CP-SOFTWARE-CLUSTER">/SoftwareClusters/SoftwareClusterA</REQUESTOR-REF>
        </REQUESTOR-REFS>
        <TIMING-DESCRIPTION-REF DEST="TD-EVENT-COMPLEX">/TimingExtensions/SoftwareClusterHostTiming/
          Dispatcher10msEntry</TIMING-DESCRIPTION-REF>
      </TD-CP-SOFTWARE-CLUSTER-MAPPING>
      <TD-CP-SOFTWARE-CLUSTER-MAPPING>
        <SHORT-NAME>AToLet5msMapping</SHORT-NAME>
        <PROVIDER-REF DEST="CP-SOFTWARE-CLUSTER">/SoftwareClusters/SoftwareClusterHost</PROVIDER-REF>
        <REQUESTOR-REFS>
          <REQUESTOR-REF DEST="CP-SOFTWARE-CLUSTER">/SoftwareClusters/SoftwareClusterA</REQUESTOR-REF>
        </REQUESTOR-REFS>
        <TIMING-DESCRIPTION-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/TimingExtensions/SoftwareClusterHostTiming/
          Let5msInterval</TIMING-DESCRIPTION-REF>
      </TD-CP-SOFTWARE-CLUSTER-MAPPING>
    </TD-CP-SOFTWARE-CLUSTER-TO-TD-MAPPINGS>
  </TD-CP-SOFTWARE-CLUSTER-MAPPING-SET>
</ELEMENTS>
<AR-PACKAGE>
  <AR-PACKAGE>
    <SHORT-NAME>BtoHost</SHORT-NAME>
  </AR-PACKAGE>
</AR-PACKAGES>
</AR-PACKAGE>
<AR-PACKAGE UUID="0fddb17c-de73-447d-a0a6-85b85f5d2449">
  <SHORT-NAME>TimingExtensions</SHORT-NAME>
  <ELEMENTS>
    <SYSTEM-TIMING>
      <SHORT-NAME>SoftwareClusterATiming</SHORT-NAME>
      <TIMING-REQUIREMENTS>
        <EXECUTION-ORDER-CONSTRAINT>
          <SHORT-NAME>eocDispatcher10ms</SHORT-NAME>
          <IGNORE-ORDER-ALLOWED>false</IGNORE-ORDER-ALLOWED>
          <IS-EVENT>true</IS-EVENT>
        <ORDERED-ELEMENTS>
          <EOC-EXECUTABLE-ENTITY-REF-GROUP>
            <SHORT-NAME>eoceergDispatcher10ms</SHORT-NAME>
            <NESTED-ELEMENT-REFS>
              <NESTED-ELEMENT-REF DEST="EOC-EVENT-REF">/TimingExtensions/SoftwareClusterATiming/eocDispatcher10ms/
                eocerRe1ASWCT05</NESTED-ELEMENT-REF>
              <NESTED-ELEMENT-REF DEST="EOC-EVENT-REF">/TimingExtensions/SoftwareClusterATiming/eocDispatcher10ms/
                eocerRe1ASWCT02</NESTED-ELEMENT-REF>
              <NESTED-ELEMENT-REF DEST="EOC-EVENT-REF">/TimingExtensions/SoftwareClusterATiming/eocDispatcher10ms/
                eocerRe2ASWCT15</NESTED-ELEMENT-REF>
              <NESTED-ELEMENT-REF DEST="EOC-EVENT-REF">/TimingExtensions/SoftwareClusterATiming/eocDispatcher10ms/
                eocerRe2ASWCT08</NESTED-ELEMENT-REF>
              <NESTED-ELEMENT-REF DEST="EOC-EVENT-REF">/TimingExtensions/SoftwareClusterATiming/eocDispatcher10ms/
                eocerRe1ASWCT13</NESTED-ELEMENT-REF>
              <NESTED-ELEMENT-REF DEST="EOC-EVENT-REF">/TimingExtensions/SoftwareClusterATiming/eocDispatcher10ms/
                eocerRe3ASWCT09</NESTED-ELEMENT-REF>
            </NESTED-ELEMENT-REFS>
            <TRIGGERING-EVENT-REF DEST="TD-EVENT-COMPLEX">/TimingExtensions/SoftwareClusterHostTiming/
              Dispatcher10msEntry</TRIGGERING-EVENT-REF>
          </EOC-EXECUTABLE-ENTITY-REF-GROUP>
        <EOC-EVENT-REF>
          <SHORT-NAME>eoceergRe1ASWCT13</SHORT-NAME>
          <EVENT-REF DEST="TIMING-EVENT">/SwComponents/ASWCT13/ibASWCT13/teRe1ASWCT13</EVENT-REF>
        </EOC-EVENT-REF>
        <EOC-EVENT-REF>
          <SHORT-NAME>eoceergRe1ASWCT02</SHORT-NAME>
          <EVENT-REF DEST="TIMING-EVENT">/SwComponents/ASWCT02/ibASWCT02/teRe1ASWCT02</EVENT-REF>
        </EOC-EVENT-REF>
        <EOC-EVENT-REF>
          <SHORT-NAME>eoceergRe1ASWCT05</SHORT-NAME>
          <EVENT-REF DEST="TIMING-EVENT">/SwComponents/ASWCT05/ibASWCT05/teRe1ASWCT05</EVENT-REF>
        </EOC-EVENT-REF>
        <EOC-EVENT-REF>
          <SHORT-NAME>eoceergRe2ASWCT08</SHORT-NAME>
          <EVENT-REF DEST="TIMING-EVENT">/SwComponents/ASWCT08/ibASWCT08/teRe2ASWCT08</EVENT-REF>
        </EOC-EVENT-REF>
        <EOC-EVENT-REF>
          <SHORT-NAME>eoceergRe2ASWCT15</SHORT-NAME>
          <EVENT-REF DEST="TIMING-EVENT">/SwComponents/ASWCT15/ibASWCT15/teRe2ASWCT15</EVENT-REF>
        </EOC-EVENT-REF>
        <EOC-EVENT-REF>
          <SHORT-NAME>eoceergRe3ASWCT09</SHORT-NAME>
          <EVENT-REF DEST="TIMING-EVENT">/SwComponents/ASWCT09/ibASWCT09/teRe3ASWCT09</EVENT-REF>
        </EOC-EVENT-REF>
      </ORDERED-ELEMENTS>
    <EXECUTION-ORDER-CONSTRAINT>
      <EXECUTION-ORDER-CONSTRAINT>
        <SHORT-NAME>eoceergLet5ms</SHORT-NAME>
        <IGNORE-ORDER-ALLOWED>false</IGNORE-ORDER-ALLOWED>
        <IS-EVENT>true</IS-EVENT>
      <ORDERED-ELEMENTS>
        <EOC-EXECUTABLE-ENTITY-REF-GROUP>
          <SHORT-NAME>eoceergLet5ms</SHORT-NAME>
          <LET-INTERVAL-REFS>

```

```

<LET-INTERVAL-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/TimingExtensions/SoftwareClusterHostTiming/
  Let5msInterval</LET-INTERVAL-REF>
</LET-INTERVAL-REFS>
<NESTED-ELEMENT-REFS>
  <NESTED-ELEMENT-REF DEST="EOC-EVENT-REF">/TimingExtensions/SoftwareClusterATiming/eocLet5ms/
    eocerRe1ASWCT14</NESTED-ELEMENT-REF>
  <NESTED-ELEMENT-REF DEST="EOC-EVENT-REF">/TimingExtensions/SoftwareClusterATiming/eocLet5ms/
    eocerRe2ASWCT05</NESTED-ELEMENT-REF>
  <NESTED-ELEMENT-REF DEST="EOC-EVENT-REF">/TimingExtensions/SoftwareClusterATiming/eocLet5ms/
    eocerRe2ASWCT03</NESTED-ELEMENT-REF>
  <NESTED-ELEMENT-REF DEST="EOC-EVENT-REF">/TimingExtensions/SoftwareClusterATiming/eocLet5ms/
    eocerRe3ASWCT06</NESTED-ELEMENT-REF>
</NESTED-ELEMENT-REFS>
</EOC-EXECUTABLE-ENTITY-REF-GROUP>
<EOC-EVENT-REF>
  <SHORT-NAME>eocerRe1ASWCT14</SHORT-NAME>
  <EVENT-REF DEST="TIMING-EVENT">/SwComponents/ASWCT14/ibASWCT14/teRe1ASWCT14</EVENT-REF>
</EOC-EVENT-REF>
<EOC-EVENT-REF>
  <SHORT-NAME>eocerRe2ASWCT03</SHORT-NAME>
  <EVENT-REF DEST="TIMING-EVENT">/SwComponents/ASWCT03/ibASWCT03/teRe2ASWCT03</EVENT-REF>
</EOC-EVENT-REF>
<EOC-EVENT-REF>
  <SHORT-NAME>eocerRe2ASWCT05</SHORT-NAME>
  <EVENT-REF DEST="TIMING-EVENT">/SwComponents/ASWCT05/ibASWCT05/teRe2ASWCT05</EVENT-REF>
</EOC-EVENT-REF>
<EOC-EVENT-REF>
  <SHORT-NAME>eocerRe3ASWCT06</SHORT-NAME>
  <EVENT-REF DEST="TIMING-EVENT">/SwComponents/ASWCT06/ibASWCT06/teRe3ASWCT06</EVENT-REF>
</EOC-EVENT-REF>
</ORDERED-ELEMENTS>
<PERMIT-MULTIPLE-REFERENCES-TO-EE>false</PERMIT-MULTIPLE-REFERENCES-TO-EE>
</EXECUTION-ORDER-CONSTRAINT>
</TIMING-REQUIREMENTS>
<SYSTEM-REF DEST="SYSTEM">/System/SoftwareClusterA</SYSTEM-REF>
</SYSTEM-TIMING>
<SYSTEM-TIMING S="">
  <SHORT-NAME>SoftwareClusterBTiming</SHORT-NAME>
  <SYSTEM-REF DEST="SYSTEM">/System/SoftwareClusterB</SYSTEM-REF>
</SYSTEM-TIMING>
<SYSTEM-TIMING UUID="">
  <SHORT-NAME>SoftwareClusterHostTiming</SHORT-NAME>
  <TIMING-DESCRIPTIONS>
    <TIMING-DESCRIPTION-EVENT-CHAIN>
      <SHORT-NAME>Dispatcher10ms</SHORT-NAME>
      <STIMULUS-REF DEST="TD-EVENT-COMPLEX">/TimingExtensions/SoftwareClusterHostTiming/Dispatcher10msEntry</STIMULUS-REF>
      <RESPONSE-REF DEST="TD-EVENT-COMPLEX">/TimingExtensions/SoftwareClusterHostTiming/Dispatcher10msExit</RESPONSE-REF>
      <SEGMENT-REFS>
        <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/TimingExtensions/SoftwareClusterHostTiming/Dispatcher10ms</SEGMENT-REF>
      </SEGMENT-REFS>
    </TIMING-DESCRIPTION-EVENT-CHAIN>
    <TD-EVENT-COMPLEX>
      <SHORT-NAME>Dispatcher10msEntry</SHORT-NAME>
      <CATEGORY>DISPATCH_ENTRY_POINT</CATEGORY>
    </TD-EVENT-COMPLEX>
    <TD-EVENT-COMPLEX>
      <SHORT-NAME>Dispatcher10msExit</SHORT-NAME>
      <CATEGORY>DISPATCH_EXIT_POINT</CATEGORY>
    </TD-EVENT-COMPLEX>
    <TIMING-DESCRIPTION-EVENT-CHAIN>
      <SHORT-NAME>Let5msInterval</SHORT-NAME>
      <CATEGORY>LET_INTERVAL</CATEGORY>
      <STIMULUS-REF DEST="TD-EVENT-COMPLEX">/TimingExtensions/SoftwareClusterHostTiming/Let5msRelease</STIMULUS-REF>
      <RESPONSE-REF DEST="TD-EVENT-COMPLEX">/TimingExtensions/SoftwareClusterHostTiming/Let5msTerminate</RESPONSE-REF>
      <SEGMENT-REFS>
        <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/TimingExtensions/SoftwareClusterHostTiming/Let5msInterval</SEGMENT-REF>
      </SEGMENT-REFS>
    </TIMING-DESCRIPTION-EVENT-CHAIN>
    <TD-EVENT-COMPLEX>
      <SHORT-NAME>Let5msRelease</SHORT-NAME>
      <CATEGORY>LET_RELEASE</CATEGORY>
    </TD-EVENT-COMPLEX>
    <TD-EVENT-COMPLEX>
      <SHORT-NAME>Let5msTerminate</SHORT-NAME>
      <CATEGORY>LET_TERMINATE</CATEGORY>
    </TD-EVENT-COMPLEX>
  </TIMING-DESCRIPTIONS>
  <TIMING-REQUIREMENTS>
    <LATENCY-TIMING-CONSTRAINT>
      <SHORT-NAME>Dispatcher10msMaxExecutionTime</SHORT-NAME>
      <LATENCY-CONSTRAINT-TYPE>REACTION</LATENCY-CONSTRAINT-TYPE>
      <SCOPE-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/TimingExtensions/SoftwareClusterHostTiming/Dispatcher10ms</SCOPE-REF>
      <MINIMUM>
        <CSE-CODE>2</CSE-CODE>
        <CSE-CODE-FACTOR>0</CSE-CODE-FACTOR>
      </MINIMUM>
      <MAXIMUM>
        <CSE-CODE>2</CSE-CODE>
        <CSE-CODE-FACTOR>5</CSE-CODE-FACTOR>
      </MAXIMUM>
    </LATENCY-TIMING-CONSTRAINT>
  </TIMING-REQUIREMENTS>

```

```

</LATENCY-TIMING-CONSTRAINT>
<PERIODIC-EVENT-TRIGGERING>
  <SHORT-NAME>Dispatcher10msPetc</SHORT-NAME>
  <EVENT-REF DEST="TD-EVENT-COMPLEX">/TimingExtensions/SoftwareClusterHostTiming/Dispatcher10msEntry</EVENT-REF>
  <MINIMUM-INTER-ARRIVAL-TIME>
    <CSE-CODE>3</CSE-CODE>
    <CSE-CODE-FACTOR>10</CSE-CODE-FACTOR>
  </MINIMUM-INTER-ARRIVAL-TIME>
  <JITTER>
    <CSE-CODE>3</CSE-CODE>
    <CSE-CODE-FACTOR>1</CSE-CODE-FACTOR>
  </JITTER>
  <PERIOD>
    <CSE-CODE>3</CSE-CODE>
    <CSE-CODE-FACTOR>10</CSE-CODE-FACTOR>
  </PERIOD>
</PERIODIC-EVENT-TRIGGERING>
<LATENCY-TIMING-CONSTRAINT>
  <SHORT-NAME>Let5msLatency</SHORT-NAME>
  <LATENCY-CONSTRAINT-TYPE>REACTION</LATENCY-CONSTRAINT-TYPE>
  <MINIMUM>
    <CSE-CODE>3</CSE-CODE>
    <CSE-CODE-FACTOR>5</CSE-CODE-FACTOR>
  </MINIMUM>
  <MAXIMUM>
    <CSE-CODE>3</CSE-CODE>
    <CSE-CODE-FACTOR>5</CSE-CODE-FACTOR>
  </MAXIMUM>
</LATENCY-TIMING-CONSTRAINT>
<PERIODIC-EVENT-TRIGGERING>
  <SHORT-NAME>Let5msPetc</SHORT-NAME>
  <EVENT-REF DEST="TD-EVENT-COMPLEX">/TimingExtensions/SoftwareClusterHostTiming/Let5msRelease</EVENT-REF>
  <MINIMUM-INTER-ARRIVAL-TIME>
    <CSE-CODE>3</CSE-CODE>
    <CSE-CODE-FACTOR>5</CSE-CODE-FACTOR>
  </MINIMUM-INTER-ARRIVAL-TIME>
  <JITTER>
    <CSE-CODE>3</CSE-CODE>
    <CSE-CODE-FACTOR>0</CSE-CODE-FACTOR>
  </JITTER>
  <PERIOD>
    <CSE-CODE>3</CSE-CODE>
    <CSE-CODE-FACTOR>5</CSE-CODE-FACTOR>
  </PERIOD>
</PERIODIC-EVENT-TRIGGERING>
</TIMING-REQUIREMENTS>
<SYSTEM-REF DEST="SYSTEM">/System/SoftwareClusterHost</SYSTEM-REF>
</SYSTEM-TIMING>
</ELEMENTS>
</AR-PACKAGE>
</AR-PACKAGES>
</AUTOSAR>
  
```

Listing C.11: Timing Extensions - Software Cluster Dispatcher Logical Execution Time

C.4 System Level Logical Execution Time

C.4.1 System Level Logical Execution Time

```

<?xml version="1.0" encoding="utf-8"?>
<!--
  Purpose: Example ARXML listings for use in TPS_AbstractPlatformDescription
  Date: First version 2020-August
  Author: B.R.
-->
<AUTOSAR xmlns="http://autosar.org/schema/r4.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://autosar.org/schema/r4.0_AUTOSAR_00053.xsd">
  <AR-PACKAGES>
    <AR-PACKAGE>
      <SHORT-NAME>SystemLevelLogicalExecutionTime</SHORT-NAME>
      <AR-PACKAGES>
        <AR-PACKAGE>
          <SHORT-NAME>Types</SHORT-NAME>
          <ELEMENTS>
            <APPLICATION-PRIMITIVE-DATA-TYPE>
              <SHORT-NAME>AppDataTypel</SHORT-NAME>
              <CATEGORY>VALUE</CATEGORY>
            </APPLICATION-PRIMITIVE-DATA-TYPE>
          </ELEMENTS>
        </AR-PACKAGE>
        <AR-PACKAGE>
          <SHORT-NAME>Interfaces</SHORT-NAME>
          <ELEMENTS>
            <SENDER-RECEIVER-INTERFACE>
              <SHORT-NAME>SRIInterface1</SHORT-NAME>
              <DATA-ELEMENTS>
                <VARIABLE-DATA-PROTOTYPE>
                  <SHORT-NAME>vdp_X</SHORT-NAME>
                </VARIABLE-DATA-PROTOTYPE>
              </DATA-ELEMENTS>
            </SENDER-RECEIVER-INTERFACE>
          </ELEMENTS>
        </AR-PACKAGE>
      </AR-PACKAGES>
    </AR-PACKAGE>
  </AR-PACKAGES>
</AUTOSAR>
  
```

```

<TYPE-TREF DEST="APPLICATION-PRIMITIVE-DATA-TYPE">/SystemLevelLogicalExecutionTime/Types/AppDataType1</TYPE-TREF>
<VARIABLE-DATA-PROTOTYPE>
<VARIABLE-DATA-PROTOTYPE>
<SHORT-NAME>vdp_Y</SHORT-NAME>
<TYPE-TREF DEST="APPLICATION-DATA-TYPE">/SystemLevelLogicalExecutionTime/Types/AppDataType1</TYPE-TREF>
</VARIABLE-DATA-PROTOTYPE>
</DATA-ELEMENTS>
</SENDER-RECEIVER-INTERFACE>
</ELEMENTS>
</AR-PACKAGE>
<AR-PACKAGE>
<SHORT-NAME>Components</SHORT-NAME>
<ELEMENTS>
<ADAPTIVE-APPLICATION-SW-COMPONENT-TYPE>
<SHORT-NAME>ApAppSwc1</SHORT-NAME>
<PORTS>
<P-PORT-PROTOTYPE>
<SHORT-NAME>PPort1</SHORT-NAME>
<PROVIDED-INTERFACE-TREF DEST="SENDER-RECEIVER-INTERFACE">/SystemLevelLogicalExecutionTime/Interfaces/SRInterface1</PROVIDED-INTERFACE-TREF>
</P-PORT-PROTOTYPE>
<P-PORT-PROTOTYPE>
<SHORT-NAME>PPort2</SHORT-NAME>
<PROVIDED-INTERFACE-TREF DEST="SENDER-RECEIVER-INTERFACE">/SystemLevelLogicalExecutionTime/Interfaces/SRInterface1</PROVIDED-INTERFACE-TREF>
</P-PORT-PROTOTYPE>
</PORTS>
<ADAPTIVE-APPLICATION-SW-COMPONENT-TYPE>
<APPLICATION-SW-COMPONENT-TYPE>
<SHORT-NAME>CpAppSwc1</SHORT-NAME>
<PORTS>
<R-PORT-PROTOTYPE>
<SHORT-NAME>RPort2</SHORT-NAME>
<REQUIRED-INTERFACE-TREF DEST="SENDER-RECEIVER-INTERFACE">/SystemLevelLogicalExecutionTime/Interfaces/SRInterface1</REQUIRED-INTERFACE-TREF>
</R-PORT-PROTOTYPE>
<R-PORT-PROTOTYPE>
<SHORT-NAME>RPort1</SHORT-NAME>
<REQUIRED-INTERFACE-TREF DEST="SENDER-RECEIVER-INTERFACE">/SystemLevelLogicalExecutionTime/Interfaces/SRInterface1</REQUIRED-INTERFACE-TREF>
</R-PORT-PROTOTYPE>
</PORTS>
<APPLICATION-SW-COMPONENT-TYPE>
<COMPOSITION-SW-COMPONENT-TYPE>
<SHORT-NAME>CpCompSwc1</SHORT-NAME>
<COMPONENTS>
<SW-COMPONENT-PROTOTYPE>
<SHORT-NAME>CpAppSwc1</SHORT-NAME>
<TYPE-TREF DEST="APPLICATION-SW-COMPONENT-TYPE">/SystemLevelLogicalExecutionTime/Components/CpAppSwc1</TYPE-TREF>
</SW-COMPONENT-PROTOTYPE>
</COMPONENTS>
</COMPOSITION-SW-COMPONENT-TYPE>
<COMPOSITION-SW-COMPONENT-TYPE>
<SHORT-NAME>ApCompSwc1</SHORT-NAME>
<COMPONENTS>
<SW-COMPONENT-PROTOTYPE>
<SHORT-NAME>ApAppSwc1</SHORT-NAME>
</SW-COMPONENT-PROTOTYPE>
</COMPONENTS>
</COMPOSITION-SW-COMPONENT-TYPE>
</ELEMENTS>
</AR-PACKAGE>
<AR-PACKAGE>
<SHORT-NAME>TimingExtensions</SHORT-NAME>
<ELEMENTS>
<SYSTEM-TIMING>
<SHORT-NAME>SystemTiming</SHORT-NAME>
<TIMING-CLOCKS>
<TDLET-ZONE-CLOCK>
<SHORT-NAME>ZoneClock0</SHORT-NAME>
<ACCURACY-INT>
<CSE-CODE>0</CSE-CODE>
<CSE-CODE-FACTOR>30</CSE-CODE-FACTOR>
</ACCURACY-INT>
</TDLET-ZONE-CLOCK>
<TDLET-ZONE-CLOCK>
<SHORT-NAME>ZoneClock1</SHORT-NAME>
</TDLET-ZONE-CLOCK>
<TDLET-ZONE-CLOCK>
<SHORT-NAME>ZoneClock2</SHORT-NAME>
</TDLET-ZONE-CLOCK>
</TIMING-CLOCKS>
<TIMING-CLOCK-SYNC-ACCURACYS>
<TIMING-CLOCK-SYNC-ACCURACY>
<SHORT-NAME>SyncAccuracy1</SHORT-NAME>
<ACCURACY>
<CSE-CODE>0</CSE-CODE>
<CSE-CODE-FACTOR>10</CSE-CODE-FACTOR>
</ACCURACY>
<LOWER-REF DEST="TDLET-ZONE-CLOCK">/SystemLevelLogicalExecutionTime/TimingExtensions/SystemTiming/ZoneClock1</LOWER-REF>
<UPPER-REF DEST="TDLET-ZONE-CLOCK">/SystemLevelLogicalExecutionTime/TimingExtensions/SystemTiming/ZoneClock0</UPPER-REF>
</ELEMENTS>

```

```

</TIMING-CLOCK-SYNC-ACCURACY>
<TIMING-CLOCK-SYNC-ACCURACY>
    <SHORT-NAME>SyncAccuracy2</SHORT-NAME>
    <ACCURACY>
        <CSE-CODE>0</CSE-CODE>
        <CSE-CODE-FACTOR>5</CSE-CODE-FACTOR>
    </ACCURACY>
    <LOWER-REF DEST="TDLET-ZONE-CLOCK">/SystemLevelLogicalExecutionTime/TimingExtensions/SystemTiming/ZoneClock2
    </LOWER-REF>
    <UPPER-REF DEST="TDLET-ZONE-CLOCK">/SystemLevelLogicalExecutionTime/TimingExtensions/SystemTiming/ZoneClock0
    </UPPER-REF>
</TIMING-CLOCK-SYNC-ACCURACY>
</TIMING-CLOCK-SYNC-ACCURACYS>
<!-- start example slet tds - line 48 -->
<TIMING-DESCRIPTIONS>
    <TD-EVENT-VARIABLE-DATA-PROTOTYPE>
        <SHORT-NAME>tdevdp_XinRPort1</SHORT-NAME>
        <CLOCK-REFERENCE-REF DEST="TDLET-ZONE-CLOCK">/SystemLevelLogicalExecutionTime/TimingExtensions/SystemTiming/
            ZoneClock0</CLOCK-REFERENCE-REF>
        <PORT-PROTOTYPE-IREF>
            <R-PORT-IN-COMPOSITION-INSTANCE-REF>
                <CONTEXT-COMPONENT-REF DEST="SW-COMPONENT-PROTOTYPE">/SystemLevelLogicalExecutionTime/Components/
                    CpCompSwc1/CpAppSwc1</CONTEXT-COMPONENT-REF>
                <TARGET-R-PORT-REF DEST="R-PORT-PROTOTYPE">/SystemLevelLogicalExecutionTime/Components/CpAppSwc1/RPort1<
                    /TARGET-R-PORT-REF>
            </R-PORT-IN-COMPOSITION-INSTANCE-REF>
        </PORT-PROTOTYPE-IREF>
        <DATA-ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE">/SystemLevelLogicalExecutionTime/Interfaces/SRInterface1/
            vdp_X</DATA-ELEMENT-REF>
    </TD-EVENT-VARIABLE-DATA-PROTOTYPE>
    <TD-EVENT-VARIABLE-DATA-PROTOTYPE>
        <SHORT-NAME>tdevdp_XinRPort2</SHORT-NAME>
        <CLOCK-REFERENCE-REF DEST="TDLET-ZONE-CLOCK">/SystemLevelLogicalExecutionTime/TimingExtensions/SystemTiming/
            ZoneClock0</CLOCK-REFERENCE-REF>
        <PORT-PROTOTYPE-IREF>
            <R-PORT-IN-COMPOSITION-INSTANCE-REF>
                <CONTEXT-COMPONENT-REF DEST="SW-COMPONENT-PROTOTYPE">/SystemLevelLogicalExecutionTime/Components/
                    CpCompSwc1/CpAppSwc1</CONTEXT-COMPONENT-REF>
                <TARGET-R-PORT-REF DEST="R-PORT-PROTOTYPE">/SystemLevelLogicalExecutionTime/Components/CpAppSwc1/RPort2<
                    /TARGET-R-PORT-REF>
            </R-PORT-IN-COMPOSITION-INSTANCE-REF>
        </PORT-PROTOTYPE-IREF>
        <DATA-ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE">/SystemLevelLogicalExecutionTime/Interfaces/SRInterface1/
            vdp_X</DATA-ELEMENT-REF>
    </TD-EVENT-VARIABLE-DATA-PROTOTYPE>
    <TD-EVENT-VARIABLE-DATA-PROTOTYPE>
        <SHORT-NAME>tdevdp_XinPPort1</SHORT-NAME>
        <CLOCK-REFERENCE-REF DEST="TDLET-ZONE-CLOCK">/SystemLevelLogicalExecutionTime/TimingExtensions/SystemTiming/
            ZoneClock2</CLOCK-REFERENCE-REF>
        <PORT-PROTOTYPE-IREF>
            <P-PORT-IN-COMPOSITION-INSTANCE-REF>
                <CONTEXT-COMPONENT-REF DEST="SW-COMPONENT-PROTOTYPE">/SystemLevelLogicalExecutionTime/Components/
                    ApCompSwc1/ApAppSwc1</CONTEXT-COMPONENT-REF>
                <TARGET-P-PORT-REF DEST="P-PORT-PROTOTYPE">/SystemLevelLogicalExecutionTime/Components/ApAppSwc1/PPort1<
                    /TARGET-P-PORT-REF>
            </P-PORT-IN-COMPOSITION-INSTANCE-REF>
        </PORT-PROTOTYPE-IREF>
        <DATA-ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE">/SystemLevelLogicalExecutionTime/Interfaces/SRInterface1/
            vdp_X</DATA-ELEMENT-REF>
    </TD-EVENT-VARIABLE-DATA-PROTOTYPE>
    <TD-EVENT-VARIABLE-DATA-PROTOTYPE>
        <SHORT-NAME>tdevdp_YinPPort2</SHORT-NAME>
        <CLOCK-REFERENCE-REF DEST="TDLET-ZONE-CLOCK">/SystemLevelLogicalExecutionTime/TimingExtensions/SystemTiming/
            ZoneClock2</CLOCK-REFERENCE-REF>
        <PORT-PROTOTYPE-IREF>
            <P-PORT-IN-COMPOSITION-INSTANCE-REF>
                <CONTEXT-COMPONENT-REF DEST="SW-COMPONENT-PROTOTYPE">/SystemLevelLogicalExecutionTime/Components/
                    ApCompSwc1/ApAppSwc1</CONTEXT-COMPONENT-REF>
                <TARGET-P-PORT-REF DEST="P-PORT-PROTOTYPE">/SystemLevelLogicalExecutionTime/Components/ApAppSwc1/PPort2<
                    /TARGET-P-PORT-REF>
            </P-PORT-IN-COMPOSITION-INSTANCE-REF>
        </PORT-PROTOTYPE-IREF>
        <DATA-ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE">/SystemLevelLogicalExecutionTime/Interfaces/SRInterface1/
            vdp_Y</DATA-ELEMENT-REF>
    </TD-EVENT-VARIABLE-DATA-PROTOTYPE>
    <TD-EVENT-VFB-PORT-GROUP>
        <SHORT-NAME>tdevfbpg_RPortGroup1</SHORT-NAME>
        <TD-EVENT-VFB-PORT-REFS>
            <TD-EVENT-VFB-PORT-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/SystemLevelLogicalExecutionTime/
                TimingExtensions/SystemTiming/tdevdp_XinRPort1</TD-EVENT-VFB-PORT-REF>
            <TD-EVENT-VFB-PORT-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/SystemLevelLogicalExecutionTime/
                TimingExtensions/SystemTiming/tdevdp_XinRPort2</TD-EVENT-VFB-PORT-REF>
        </TD-EVENT-VFB-PORT-REFS>
    </TD-EVENT-VFB-PORT-GROUP>
    <TD-EVENT-VFB-PORT-GROUP>
        <SHORT-NAME>tdevfbpg_PPortGroup1</SHORT-NAME>
        <TD-EVENT-VFB-PORT-REFS>
            <TD-EVENT-VFB-PORT-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/SystemLevelLogicalExecutionTime/
                TimingExtensions/SystemTiming/tdevdp_XinPPort1</TD-EVENT-VFB-PORT-REF>
            <TD-EVENT-VFB-PORT-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/SystemLevelLogicalExecutionTime/
                TimingExtensions/SystemTiming/tdevdp_YinPPort2</TD-EVENT-VFB-PORT-REF>
        </TD-EVENT-VFB-PORT-REFS>
    </TD-EVENT-VFB-PORT-GROUP>
    <TIMING-DESCRIPTION-EVENT-CHAIN>
        <SHORT-NAME>TimingDescriptionChain1</SHORT-NAME>
    </TIMING-DESCRIPTION-EVENT-CHAIN>

```

```

<CATEGORY>SL_Let_INTERVAL</CATEGORY>
<STIMULUS-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/SystemLevelLogicalExecutionTime/TimingExtensions/
  SystemTiming/tdevdp_XinPPort1</STIMULUS-REF>
<RESPONSE-REF DEST="TD-EVENT-VFB-PORT-GROUP">/SystemLevelLogicalExecutionTime/TimingExtensions/SystemTiming/
  tdevfbpg_RPortGroup1</RESPONSE-REF>
</TIMING-DESCRIPTION-EVENT-CHAIN>
<TIMING-DESCRIPTION-EVENT-CHAIN>
  <SHORT-NAME>TimingDescriptionChain3</SHORT-NAME>
  <CATEGORY>SL_Let_INTERVAL</CATEGORY>
  <STIMULUS-REF DEST="TD-EVENT-VFB-PORT-GROUP">/SystemLevelLogicalExecutionTime/TimingExtensions/SystemTiming/
    tdevfbpg_RPortGroup1</STIMULUS-REF>
  <RESPONSE-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/SystemLevelLogicalExecutionTime/TimingExtensions/
    SystemTiming/tdevdp_XinRPort1</RESPONSE-REF>
</TIMING-DESCRIPTION-EVENT-CHAIN>
</TIMING-DESCRIPTIONS>
<!-- end example silet tds - line 72 -->
<!-- start example silet duration/recurrence - line 73 -->
<TIMING-REQUIREMENTS>
  <PERIODIC-EVENT-TRIGGERING>
    <SHORT-NAME>PeriodicEventTriggeringConstraintS1LetRecurrence1</SHORT-NAME>
    <MINIMUM-INTER-ARRIVAL-TIME>
      <CSE-CODE>3</CSE-CODE>
      <CSE-CODE-FACTOR>20</CSE-CODE-FACTOR>
    </MINIMUM-INTER-ARRIVAL-TIME>
    <JITTER>
      <CSE-CODE>3</CSE-CODE>
      <CSE-CODE-FACTOR>0</CSE-CODE-FACTOR>
    </JITTER>
    <PERIOD>
      <CSE-CODE>3</CSE-CODE>
      <CSE-CODE-FACTOR>20</CSE-CODE-FACTOR>
    </PERIOD>
  </PERIODIC-EVENT-TRIGGERING>
  <LATENCY-TIMING-CONSTRAINT>
    <SHORT-NAME>LatencyTimingConstraintS1LetDuration1</SHORT-NAME>
    <LATENCY-CONSTRAINT-TYPE>REACTION</LATENCY-CONSTRAINT-TYPE>
    <SCOPE-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/SystemLevelLogicalExecutionTime/TimingExtensions/
      SystemTiming/TimingDescriptionChain1</SCOPE-REF>
    <MAXIMUM>
      <CSE-CODE>3</CSE-CODE>
      <CSE-CODE-FACTOR>400</CSE-CODE-FACTOR>
    </MAXIMUM>
  </LATENCY-TIMING-CONSTRAINT>
  </TIMING-REQUIREMENTS>
  <!-- end example silet duration/recurrence - line 101 -->
</SYSTEM-TIMING>
</ELEMENTS>
</AR-PACKAGE>
</AR-PACKAGES>
</AR-PACKAGE>
</AR-PACKAGES>
</AUTOSAR>
  
```

Listing C.12: Timing Extensions - System Level Logical Execution Time

C.5 Timing Description Event Occurrence

C.5.1 Timing Description Event Occurrence - Filter Expression

```

1  // Condition 1
2  TIMEX_occurs( reference( "TD-EVENT-VARIABLE-DATA-PROTOTYPE:/Example/Expression/E1" ) )
3  || TIMEX_occurs( reference( "TD-EVENT-VARIABLE-DATA-PROTOTYPE:/Example/Expression/E2" ) )
4  // Condition 2
5  && TIMEX_value( reference( "AUTOSAR-VARIABLE-INSTANCE:/Example/Expression/ECX/avi_DE3" ) ) > 3
6  // Condition 3
7  && abs( TIMEX_timeSinceLastOccurrence( reference( "TD-EVENT-VARIABLE-DATA-PROTOTYPE:/Example/Expression/E1" ) ) -
8    TIMEX_timeSinceLastOccurrence( reference( "TD-EVENT-VARIABLE-DATA-PROTOTYPE:/Example/Expression/E2" ) ) ) <= 0.0005
  
```

Listing C.13: Event Occurrence Filter

C.5.2 Timing Description Event Occurrence - Filter Expression in ARXML

```

<?xml version="1.0" encoding="UTF-8"?>
<AUTOSAR xmlns="http://autosar.org/schema/r4.0" xmlns: xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="
  http://autosar.org/schema/r4.0/AUTOSAR_00052.xsd">
<AR-PACKAGES>
  <AR-PACKAGE>
    <SHORT-NAME>Example</SHORT-NAME>
    <ELEMENTS>
      <VFB-TIMING>
        <SHORT-NAME>Expression</SHORT-NAME>
        <SHORT-NAME-PATTERN />
        <TIMING-DESCRIPTIONS>
          <TD-EVENT-VARIABLE-DATA-PROTOTYPE>
  
```

```

<SHORT-NAME>E1</SHORT-NAME>
<PORT-REF DEST="R-PORT-PROTOTYPE">/Example/Swcl/RequiredPort</PORT-REF>
<DATA-ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE">/Example/SenderReceiverInterface1/DE1</DATA-ELEMENT-REF>
<TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>VARIABLE-DATA-PROTOTYPE-RECEIVED</TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>
</TD-EVENT-VARIABLE-DATA-PROTOTYPE>
<TD-EVENT-VARIABLE-DATA-PROTOTYPE>
    <SHORT-NAME>E2</SHORT-NAME>
    <PORT-REF DEST="R-PORT-PROTOTYPE">/Example/Swcl/RequiredPort</PORT-REF>
    <DATA-ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE">/Example/SenderReceiverInterface1/DE2</DATA-ELEMENT-REF>
    <TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>VARIABLE-DATA-PROTOTYPE-RECEIVED</TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>
</TD-EVENT-VARIABLE-DATA-PROTOTYPE>
<TD-EVENT-VARIABLE-DATA-PROTOTYPE>
    <SHORT-NAME>E3</SHORT-NAME>
    <PORT-REF DEST="R-PORT-PROTOTYPE">/Example/Swcl/RequiredPort</PORT-REF>
    <DATA-ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE">/Example/SenderReceiverInterface1/DE3</DATA-ELEMENT-REF>
    <TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>VARIABLE-DATA-PROTOTYPE-RECEIVED</TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>
</TD-EVENT-VARIABLE-DATA-PROTOTYPE>
<TD-EVENT-COMPLEX>
    <SHORT-NAME>ECX</SHORT-NAME>
    <OCCURRENCE-EXPRESSION>
        <FORMULA>
            ((TIMEX_occurs(<EVENT-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE"> /Example/Expression/E1<EVENT-REF>)
            || TIMEX_occurs(<EVENT-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE"> /Example/Expression/E2<EVENT-REF>))
            &amp; &amp;
            TIMEX_value(<VARIABLE-REF DEST="AUTOSAR-VARIABLE-INSTANCE"> /Example/Expression/ECX/avi_DE3 <VARIABLE-REF>)
            &amp; &amp;
            abs(TIMEX_timeSinceLastOccurrence(<EVENT-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE"> /Example/Expression/E1<EVENT-REF>)
            - TIMEX_timeSinceLastOccurrence(<EVENT-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE"> /Example/Expression/E2<EVENT-REF>))
            &lt;= 0.0005)
        </FORMULA>
    <VARIABLES>
        <AUTOSAR-VARIABLE-INSTANCE>
            <SHORT-NAME>avi_DE3</SHORT-NAME>
            <VARIABLE-INSTANCE-IREF>
                <CONTEXT-PORT-PROTOTYPE-REF DEST="P-PORT-PROTOTYPE">/Example/Swcl/ProvidedPort</CONTEXT-PORT-PROTOTYPE-REF>
                <TARGET-DATA-PROTOTYPE-REF DEST="VARIABLE-DATA-PROTOTYPE">/Example/SenderReceiverInterface1/DE3</TARGET-DATA-PROTOTYPE-REF>
            </VARIABLE-INSTANCE-IREF>
            <AUTOSAR-VARIABLE-INSTANCE>
            </VARIABLES>
            <OCCURRENCE-EXPRESSION>
        </TD-EVENT-COMPLEX>
    </TIMING-DESCRIPTIONS>
    <COMPONENT-REF DEST="APPLICATION-SW-COMPONENT-TYPE">/Example/Swcl</COMPONENT-REF>
</VFB-TIMING>
<SENDER-RECEIVER-INTERFACE>
    <SHORT-NAME>SenderReceiverInterface1</SHORT-NAME>
    <DATA-ELEMENTS>
        <VARIABLE-DATA-PROTOTYPE>
            <SHORT-NAME>DE1</SHORT-NAME>
        </VARIABLE-DATA-PROTOTYPE>
        <VARIABLE-DATA-PROTOTYPE>
            <SHORT-NAME>DE2</SHORT-NAME>
        </VARIABLE-DATA-PROTOTYPE>
        <VARIABLE-DATA-PROTOTYPE>
            <SHORT-NAME>DE3</SHORT-NAME>
        </VARIABLE-DATA-PROTOTYPE>
    </DATA-ELEMENTS>
</SENDER-RECEIVER-INTERFACE>
<APPLICATION-SW-COMPONENT-TYPE>
    <SHORT-NAME>Swcl</SHORT-NAME>
    <PORTS>
        <P-PORT-PROTOTYPE>
            <SHORT-NAME>ProvidedPort</SHORT-NAME>
        </P-PORT-PROTOTYPE>
        <R-PORT-PROTOTYPE>
            <SHORT-NAME>RequiredPort</SHORT-NAME>
            <REQUIRED-INTERFACE-TREF DEST="SENDER-RECEIVER-INTERFACE">/Example/SenderReceiverInterface1</REQUIRED-INTERFACE-TREF>
        </R-PORT-PROTOTYPE>
    </PORTS>
    <APPLICATION-SW-COMPONENT-TYPE>
    <ELEMENTS>
    </ELEMENTS>
</APPLICATION-SW-COMPONENT-TYPE>
</AR-PACKAGE>
</AR-PACKAGES>
</AUTOSAR>

```

Listing C.14: AUTOSAR ARXML representation of the occurrence expression for the complex event "ECX"

C.5.3 Timing Description Event Occurrence - Grammar

```

ExtUnaryFuncName  :  'TIMEX_value' |
                     'TIMEX_occurs' |
                     'TIMEX_hasOccurred' |

```

```

    'TIMEX_timeSinceLastOccurrence' |
    'TIMEX_angleSinceLastOccurrence' |
    'TIMEX_modeActive'
  ;

```

Listing C.15: AUTOSAR Timing Extensions Occurrence Expression Language

C.6 Application Notes

C.6.1 Application Notes - VFB Elements

```

<AR-PACKAGE>
  <SHORT-NAME>Interfaces</SHORT-NAME>
  <ELEMENTS>
    <SENDER-RECEIVER-INTERFACE>
      <SHORT-NAME>InternalSensorData</SHORT-NAME>
      <DATA-ELEMENTS>
        <VARIABLE-DATA-PROTOTYPE>
          <SHORT-NAME>internalValueX</SHORT-NAME>
          <TYPE-TREF DEST="APPLICATION-PRIMITIVE-DATA-TYPE">/Datatypes/Float</TYPE-TREF>
        </VARIABLE-DATA-PROTOTYPE>
        <VARIABLE-DATA-PROTOTYPE>
          <SHORT-NAME>internalValueY</SHORT-NAME>
          <TYPE-TREF DEST="APPLICATION-PRIMITIVE-DATA-TYPE">/Datatypes/Float</TYPE-TREF>
        </VARIABLE-DATA-PROTOTYPE>
        <VARIABLE-DATA-PROTOTYPE>
          <SHORT-NAME>internalValueZ</SHORT-NAME>
          <TYPE-TREF DEST="APPLICATION-PRIMITIVE-DATA-TYPE">/Datatypes/Float</TYPE-TREF>
        </VARIABLE-DATA-PROTOTYPE>
      </DATA-ELEMENTS>
    </SENDER-RECEIVER-INTERFACE>
    <SENDER-RECEIVER-INTERFACE>
      <SHORT-NAME>OutputSensorData</SHORT-NAME>
      <DATA-ELEMENTS>
        <VARIABLE-DATA-PROTOTYPE>
          <SHORT-NAME>outValueX</SHORT-NAME>
          <TYPE-TREF DEST="APPLICATION-PRIMITIVE-DATA-TYPE">/Datatypes/UInt16</TYPE-TREF>
        </VARIABLE-DATA-PROTOTYPE>
        <VARIABLE-DATA-PROTOTYPE>
          <SHORT-NAME>outValueY</SHORT-NAME>
          <TYPE-TREF DEST="APPLICATION-PRIMITIVE-DATA-TYPE">/Datatypes/UInt16</TYPE-TREF>
        </VARIABLE-DATA-PROTOTYPE>
        <VARIABLE-DATA-PROTOTYPE>
          <SHORT-NAME>outValueZ</SHORT-NAME>
          <TYPE-TREF DEST="APPLICATION-PRIMITIVE-DATA-TYPE">/Datatypes/UInt16</TYPE-TREF>
        </VARIABLE-DATA-PROTOTYPE>
      </DATA-ELEMENTS>
    </SENDER-RECEIVER-INTERFACE>
  </ELEMENTS>
</AR-PACKAGE>
<AR-PACKAGE>
  <SHORT-NAME>SensorPackage</SHORT-NAME>
  <ELEMENTS>
    <SENSOR-ACTUATOR-SW-COMPONENT-TYPE>
      <SHORT-NAME>SensorComponent</SHORT-NAME>
      <PORTS>
        <P-PORT-PROTOTYPE>
          <SHORT-NAME>InternalSensorData</SHORT-NAME>
          <PROVIDED-INTERFACE-TREF DEST="SENDER-RECEIVER-INTERFACE">/Interfaces/InternalSensorData</PROVIDED-INTERFACE-TREF>
        </P-PORT-PROTOTYPE>
      </PORTS>
      <INTERNAL-BEHAVIORS>
        <SWC-INTERNAL-BEHAVIOR>
          <SHORT-NAME>SensorBehavior</SHORT-NAME>
          <RUNNABLES>
            <RUNNABLE-ENTITY>
              <SHORT-NAME>SensorRunnable</SHORT-NAME>
            </RUNNABLE-ENTITY>
          </RUNNABLES>
        </SWC-INTERNAL-BEHAVIOR>
      </INTERNAL-BEHAVIORS>
    </SENSOR-ACTUATOR-SW-COMPONENT-TYPE>
    <APPLICATION-SW-COMPONENT-TYPE>
      <SHORT-NAME>ConditioningComponent</SHORT-NAME>
      <PORTS>
        <R-PORT-PROTOTYPE>
          <SHORT-NAME>UnprocessedSensorData</SHORT-NAME>
          <REQUIRED-INTERFACE-TREF DEST="SENDER-RECEIVER-INTERFACE">/Interfaces/InternalSensorData</REQUIRED-INTERFACE-TREF>
        </R-PORT-PROTOTYPE>
        <P-PORT-PROTOTYPE>
          <SHORT-NAME>ProcessedSensorData</SHORT-NAME>
          <PROVIDED-INTERFACE-TREF DEST="SENDER-RECEIVER-INTERFACE">/Interfaces/InternalSensorData</PROVIDED-INTERFACE-TREF>
        </P-PORT-PROTOTYPE>
      </PORTS>
      <INTERNAL-BEHAVIORS>
        <SWC-INTERNAL-BEHAVIOR>

```

```

<SHORT-NAME>ConditioningBehavior</SHORT-NAME>
<RUNNABLES>
    <RUNNABLE-ENTITY>
        <SHORT-NAME>ConditioningRunnable</SHORT-NAME>
    </RUNNABLE-ENTITY>
</RUNNABLES>
</SWC-INTERNAL-BEHAVIOR>
</INTERNAL-BEHAVIORS>
</APPLICATION-SW-COMPONENT-TYPE>
<APPLICATION-SW-COMPONENT-TYPE>
    <SHORT-NAME>OutputComponent</SHORT-NAME>
    <PORTS>
        <R-PORT-PROTOTYPE>
            <SHORT-NAME>UnprocessedSensorData</SHORT-NAME>
            <REQUIRED-INTERFACE-TREF DEST="SENDER-RECEIVER-INTERFACE">/Interfaces/InternalSensorData</REQUIRED-INTERFACE-TREF>
        </R-PORT-PROTOTYPE>
        <P-PORT-PROTOTYPE>
            <SHORT-NAME>ProcessedSensorData</SHORT-NAME>
            <PROVIDED-INTERFACE-TREF DEST="SENDER-RECEIVER-INTERFACE">/Interfaces/OutputSensorData</PROVIDED-INTERFACE-TREF>
        </P-PORT-PROTOTYPE>
    </PORTS>
    <INTERNAL-BEHAVIORS>
        <SWC-INTERNAL-BEHAVIOR>
            <SHORT-NAME>OutputBehavior</SHORT-NAME>
        <RUNNABLES>
            <RUNNABLE-ENTITY>
                <SHORT-NAME>OutputRunnable</SHORT-NAME>
            </RUNNABLE-ENTITY>
        </RUNNABLES>
        </SWC-INTERNAL-BEHAVIOR>
    </INTERNAL-BEHAVIORS>
</APPLICATION-SW-COMPONENT-TYPE>
<COMPOSITION-SW-COMPONENT-TYPE>
    <SHORT-NAME>TopLevelComposition</SHORT-NAME>
    <COMPONENTS>
        <SW-COMPONENT-PROTOTYPE>
            <SHORT-NAME>sensorComponent</SHORT-NAME>
            <TYPE-TREF DEST="SENSOR-ACTUATOR-SW-COMPONENT-TYPE">/SensorPackage/SensorComponent</TYPE-TREF>
        </SW-COMPONENT-PROTOTYPE>
        <SW-COMPONENT-PROTOTYPE>
            <SHORT-NAME>conditioningComponent</SHORT-NAME>
            <TYPE-TREF DEST="APPLICATION-SW-COMPONENT-TYPE">/SensorPackage/ConditioningComponent</TYPE-TREF>
        </SW-COMPONENT-PROTOTYPE>
        <SW-COMPONENT-PROTOTYPE>
            <SHORT-NAME>outputComponent</SHORT-NAME>
            <TYPE-TREF DEST="APPLICATION-SW-COMPONENT-TYPE">/SensorPackage/OutputComponent</TYPE-TREF>
        </SW-COMPONENT-PROTOTYPE>
    </COMPONENTS>
</COMPOSITION-SW-COMPONENT-TYPE>
</ELEMENTS>
</AR-PACKAGE>
</AR-PACKAGE>

```

Listing C.16: SWC Descriptions and Interface Definitions

C.6.2 Application Notes - VFB Timing - Latency Constraints

```

<SHORT-NAME>SensorVfbTiming</SHORT-NAME>
<TIMING-DESCRIPTIONS>
    <TD-EVENT-VARIABLE-DATA-PROTOTYPE>
        <SHORT-NAME>ConditioningReceived</SHORT-NAME>
        <IS-EXTERNAL>false</IS-EXTERNAL>
        <PORT-REF DEST="R-PORT-PROTOTYPE">/SensorPackage/ConditioningComponent/UnprocessedSensorData</PORT-REF>
        <DATA-ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE">/Interfaces/InternalSensorData/internValueX</DATA-ELEMENT-REF>
        <TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>VARIABLE-DATA-PROTOTYPE-RECEIVED</TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>
    </TD-EVENT-VARIABLE-DATA-PROTOTYPE>
    <TD-EVENT-VARIABLE-DATA-PROTOTYPE>
        <SHORT-NAME>ConditioningSent</SHORT-NAME>
        <IS-EXTERNAL>false</IS-EXTERNAL>
        <PORT-REF DEST="P-PORT-PROTOTYPE">/SensorPackage/ConditioningComponent/ProcessedSensorData</PORT-REF>
        <DATA-ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE">/Interfaces/InternalSensorData/internValueX</DATA-ELEMENT-REF>
        <TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>VARIABLE-DATA-PROTOTYPE-SENT</TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>
    </TD-EVENT-VARIABLE-DATA-PROTOTYPE>
    <TD-EVENT-VARIABLE-DATA-PROTOTYPE>
        <SHORT-NAME>SensorDataProduced</SHORT-NAME>
        <IS-EXTERNAL>true</IS-EXTERNAL>
        <PORT-REF DEST="R-PORT-PROTOTYPE">/SensorPackage/ConditioningComponent/UnprocessedSensorData</PORT-REF>
        <DATA-ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE">/Interfaces/InternalSensorData/internValueX</DATA-ELEMENT-REF>
        <TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>VARIABLE-DATA-PROTOTYPE-RECEIVED</TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>
    </TD-EVENT-VARIABLE-DATA-PROTOTYPE>
    <TIMING-DESCRIPTION-EVENT-CHAIN>
        <SHORT-NAME>InputVfbChain</SHORT-NAME>
        <STIMULUS-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/TimingPackage/SensorVfbTiming/SensorDataProduced</STIMULUS-REF>
        <RESPONSE-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/TimingPackage/SensorVfbTiming/ConditioningReceived</RESPONSE-REF>
        <SEGMENT-REFS>
            <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/TimingPackage/SensorVfbTiming/InputVfbChain</SEGMENT-REF>
        </SEGMENT-REFS>
    </TIMING-DESCRIPTION-EVENT-CHAIN>

```

```

</TIMING-DESCRIPTIONS>
<TIMING-REQUIREMENTS>
    <LATENCY-TIMING-CONSTRAINT>
        <SHORT-NAME>InputVfbLatency</SHORT-NAME>
        <LATENCY-CONSTRAINT-TYPE>AGE</LATENCY-CONSTRAINT-TYPE>
        <SCOPE-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/TimingPackage/SensorVfbTiming/InputVfbChain</SCOPE-REF>
        <MINIMUM>
            <CSE-CODE>1</CSE-CODE>
            <CSE-CODE-FACTOR>40</CSE-CODE-FACTOR>
        </MINIMUM>
        <MAXIMUM>
            <CSE-CODE>1</CSE-CODE>
            <CSE-CODE-FACTOR>50</CSE-CODE-FACTOR>
        </MAXIMUM>
    </LATENCY-TIMING-CONSTRAINT>
    <EXECUTION-ORDER-CONSTRAINT>
        <SHORT-NAME>EOC1</SHORT-NAME>
        <ORDERED-ELEMENTS>
            <EOC-EXECUTABLE-ENTITY-REF>
                <SHORT-NAME>SensorRunnableRef</SHORT-NAME>
                <COMPONENT-IREF>
                    <TARGET-COMPONENT-REF DEST="SW-COMPONENT-PROTOTYPE">/SensorPackage/TopLevelComposition/sensorComponent</TARGET-COMPONENT-REF>
                </COMPONENT-IREF>
                <EXECUTABLE-REF DEST="RUNNABLE-ENTITY">/SensorPackage/SensorComponent/SensorBehavior/SensorRunnable</EXECUTABLE-REF>
            <SUCCESSOR-REFS>
                <SUCCESSOR-REF DEST="EOC-EXECUTABLE-ENTITY-REF">/TimingPackage/SensorVfbTiming/EOC1/ConditioningRunnableRef</SUCCESSOR-REF>
            </SUCCESSOR-REFS>
        </EOC-EXECUTABLE-ENTITY-REF>
        <EOC-EXECUTABLE-ENTITY-REF>
            <SHORT-NAME>ConditioningRunnableRef</SHORT-NAME>
            <COMPONENT-IREF>
                <TARGET-COMPONENT-REF DEST="SW-COMPONENT-PROTOTYPE">/SensorPackage/TopLevelComposition/conditioningComponent</TARGET-COMPONENT-REF>
            </COMPONENT-IREF>
            <EXECUTABLE-REF DEST="RUNNABLE-ENTITY">/SensorPackage/ConditioningComponent/ConditioningBehavior/ConditioningRunnable</EXECUTABLE-REF>
            <SUCCESSOR-REFS>
                <SUCCESSOR-REF DEST="EOC-EXECUTABLE-ENTITY-REF">/TimingPackage/SensorVfbTiming/EOC1/OutputRunnableRef</SUCCESSOR-REF>
            </SUCCESSOR-REFS>
        </EOC-EXECUTABLE-ENTITY-REF>
        <EOC-EXECUTABLE-ENTITY-REF>
            <SHORT-NAME>OutputRunnableRef</SHORT-NAME>
            <COMPONENT-IREF>
                <TARGET-COMPONENT-REF DEST="SW-COMPONENT-PROTOTYPE">/SensorPackage/TopLevelComposition/outputComponent</TARGET-COMPONENT-REF>
            </COMPONENT-IREF>
            <EXECUTABLE-REF DEST="RUNNABLE-ENTITY">/SensorPackage/OutputComponent/OutputBehavior/OutputRunnable</EXECUTABLE-REF>
        </EOC-EXECUTABLE-ENTITY-REF>
    </ORDERED-ELEMENTS>
    </EXECUTION-ORDER-CONSTRAINT>
</TIMING-REQUIREMENTS>
<COMPONENT-REF DEST="COMPOSITION-SW-COMPONENT-TYPE">/SensorPackage/TopLevelComposition</COMPONENT-REF>
</VFB-TIMING>
<ECU-TIMING>
    <SHORT-NAME>SensorEcuTiming</SHORT-NAME>

```

Listing C.17: LatencyConstraint and related events on VfbTiming view

C.6.3 Application Notes - VFB Timing - Execution Order Constraints

```

<ORDERED-ELEMENTS>
    <EOC-EXECUTABLE-ENTITY-REF>
        <SHORT-NAME>SensorRunnableRef</SHORT-NAME>
        <COMPONENT-IREF>
            <TARGET-COMPONENT-REF DEST="SW-COMPONENT-PROTOTYPE">/SensorPackage/TopLevelComposition/sensorComponent</TARGET-COMPONENT-REF>
        </COMPONENT-IREF>
        <EXECUTABLE-REF DEST="RUNNABLE-ENTITY">/SensorPackage/SensorComponent/SensorBehavior/SensorRunnable</EXECUTABLE-REF>
    <SUCCESSOR-REFS>
        <SUCCESSOR-REF DEST="EOC-EXECUTABLE-ENTITY-REF">/TimingPackage/SensorVfbTiming/EOC1/ConditioningRunnableRef</SUCCESSOR-REF>
    </SUCCESSOR-REFS>
</EOC-EXECUTABLE-ENTITY-REF>
<EOC-EXECUTABLE-ENTITY-REF>
    <SHORT-NAME>ConditioningRunnableRef</SHORT-NAME>
    <COMPONENT-IREF>
        <TARGET-COMPONENT-REF DEST="SW-COMPONENT-PROTOTYPE">/SensorPackage/TopLevelComposition/conditioningComponent</TARGET-COMPONENT-REF>
    </COMPONENT-IREF>
    <EXECUTABLE-REF DEST="RUNNABLE-ENTITY">/SensorPackage/ConditioningComponent/ConditioningBehavior/ConditioningRunnable</EXECUTABLE-REF>
    <SUCCESSOR-REFS>
        <SUCCESSOR-REF DEST="EOC-EXECUTABLE-ENTITY-REF">/TimingPackage/SensorVfbTiming/EOC1/OutputRunnableRef</SUCCESSOR-REF>
    </SUCCESSOR-REFS>
</EOC-EXECUTABLE-ENTITY-REF>

```

```

</EOC-EXECUTABLE-ENTITY-REF>
<EOC-EXECUTABLE-ENTITY-REF>
  <SHORT-NAME>OutputRunnableRef</SHORT-NAME>
  <COMPONENT-IREF>
    <TARGET-COMPONENT-REF DEST="SW-COMPONENT-PROTOTYPE">/SensorPackage/TopLevelComposition/outputComponent</TARGET-COMPONENT-REF>
  </COMPONENT-IREF>
  <EXECUTABLE-REF DEST="RUNNABLE-ENTITY">/SensorPackage/OutputComponent/OutputBehavior/OutputRunnable</EXECUTABLE-REF>
</EOC-EXECUTABLE-ENTITY-REF>
</ORDERED-ELEMENTS>
</EXECUTION-ORDER-CONSTRAINT>
</TIMING-REQUIREMENTS>
<COMPONENT-REF DEST="COMPOSITION-SW-COMPONENT-TYPE">/SensorPackage/TopLevelComposition</COMPONENT-REF>
  
```

Listing C.18: Execution Order Constraint for Three Runnable Entities

C.6.4 Application Notes - COM Frame Timing

```

<FRAME-REF DEST="FLEXRAY-FRAME">/SystemDescriptionPackage/SensorFrame</FRAME-REF>
<PHYSICAL-CHANNEL-REF DEST="FLEXRAY-PHYSICAL-CHANNEL">/SystemDescriptionPackage/SampleFrCluster/FrChannel110MBit
</PHYSICAL-CHANNEL-REF>
<TD-EVENT-TYPE>FRAME-TRANSMITTED-ON-BUS</TD-EVENT-TYPE>
</TD-EVENT-FRAME>
<TIMING-DESCRIPTION-EVENT-CHAIN>
  <SHORT-NAME>SensorEcuChain</SHORT-NAME>
  
```

Listing C.19: Event describing the point in time where data is sent on the bus

C.6.5 Application Notes - COM Frame Timing - Timing Description Event Chain

```

<STIMULUS-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/TimingPackage/SensorVfbTiming/ConditioningSent</STIMULUS-REF>
<RESPONSE-REF DEST="TD-EVENT-FRAME">/TimingPackage/SensorEcuTiming/DataTransmitted</RESPONSE-REF>
<SEGMENT-REFS>
  <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/TimingPackage/SensorEcuTiming/SensorEcuChain</SEGMENT-REF>
</SEGMENT-REFS>
</TIMING-DESCRIPTION-EVENT-CHAIN>
</TIMING-DESCRIPTIONS>
<TIMING-REQUIREMENTS>
  
```

Listing C.20: Event chain describing the sending path of data

C.6.6 Application Notes - COM Frame Timing - Latency Constraint

```

<SHORT-NAME>SensorEcuLatency</SHORT-NAME>
<LATENCY-CONSTRAINT-TYPE>REACTION</LATENCY-CONSTRAINT-TYPE>
<SCOPE-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/TimingPackage/SensorEcuTiming/SensorEcuChain</SCOPE-REF>
<MAXIMUM>
  <CSE-CODE>1</CSE-CODE>
  <CSE-CODE-FACTOR>400</CSE-CODE-FACTOR>
</MAXIMUM>
</LATENCY-TIMING-CONSTRAINT>
</TIMING-REQUIREMENTS>
</ECU-TIMING>
</ELEMENTS>
  
```

Listing C.21: Latency constraint prescribing the maximum latency of sending path within the ECU

C.7 Application Notes - Engine Control

C.7.1 Application Notes - Engine Control - Requirement 1

```

<VFB-TIMING>
  <SHORT-NAME>EngineControlVfbTiming1</SHORT-NAME>
  <TIMING-DESCRIPTIONS>
    <TD-EVENT-VARIABLE-DATA-PROTOTYPE>
      <SHORT-NAME>RAcceleratorPedalPositionSensorReceived</SHORT-NAME>
      <IS-EXTERNAL>false</IS-EXTERNAL>
      <PORT-REF DEST="R-PORT-PROTOTYPE">/Components/AcceleratorPedal/RAcceleratorPedalPositionSensor</PORT-REF>
      <DATA-ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE">/Interfaces/IAcceleratorPedalPositionSensor/
        AcceleratorPedalPositionSensor</DATA-ELEMENT-REF>
      <TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>VARIABLE-DATA-PROTOTYPE-RECEIVED</TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>
    </TD-EVENT-VARIABLE-DATA-PROTOTYPE>
  </TIMING-DESCRIPTIONS>
  
```

```

<TD-EVENT-VARIABLE-DATA-PROTOTYPE>
    <SHORT-NAME>PAcceleratorPedalPositionSent</SHORT-NAME>
    <IS-EXTERNAL>false</IS-EXTERNAL>
    <PORT-REF DEST="P-PORT-PROTOTYPE">/Components/AcceleratorPedal/PAcceleratorPedalPosition</PORT-REF>
    <DATA-ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE">/Interfaces/IAcceleratorPedalPosition/AcceleratorPedalPosition</DATA-ELEMENT-REF>
    <TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>VARIABLE-DATA-PROTOTYPE-SENT</TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>
</TD-EVENT-VARIABLE-DATA-PROTOTYPE>
<TD-EVENT-VARIABLE-DATA-PROTOTYPE>
    <SHORT-NAME>RAcceleratorPedalPositionReceived</SHORT-NAME>
    <IS-EXTERNAL>false</IS-EXTERNAL>
    <PORT-REF DEST="R-PORT-PROTOTYPE">/Components/AcceleratorPedalVoter/RAcceleratorPedalPosition</PORT-REF>
    <DATA-ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE">/Interfaces/IAcceleratorPedalPosition/AcceleratorPedalPosition</DATA-ELEMENT-REF>
    <TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>VARIABLE-DATA-PROTOTYPE-RECEIVED</TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>
</TD-EVENT-VARIABLE-DATA-PROTOTYPE>
<TD-EVENT-VARIABLE-DATA-PROTOTYPE>
    <SHORT-NAME>PVotedPedalPositionSent</SHORT-NAME>
    <IS-EXTERNAL>false</IS-EXTERNAL>
    <PORT-REF DEST="P-PORT-PROTOTYPE">/Components/AcceleratorPedalVoter/PVotedPedalPosition</PORT-REF>
    <DATA-ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE">/Interfaces/IVotedPedalPosition/VotedPedalPosition</DATA-ELEMENT-REF>
    <TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>VARIABLE-DATA-PROTOTYPE-SENT</TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>
</TD-EVENT-VARIABLE-DATA-PROTOTYPE>
<TD-EVENT-VARIABLE-DATA-PROTOTYPE>
    <SHORT-NAME>RVotedPedalPositionReceived</SHORT-NAME>
    <IS-EXTERNAL>false</IS-EXTERNAL>
    <PORT-REF DEST="R-PORT-PROTOTYPE">/Components/ThrottleController/RVotedPedalPosition</PORT-REF>
    <DATA-ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE">/Interfaces/IVotedPedalPosition/VotedPedalPosition</DATA-ELEMENT-REF>
    <TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>VARIABLE-DATA-PROTOTYPE-RECEIVED</TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>
</TD-EVENT-VARIABLE-DATA-PROTOTYPE>
<TD-EVENT-VARIABLE-DATA-PROTOTYPE>
    <SHORT-NAME>PUnlimitedThrottlePositionSent</SHORT-NAME>
    <IS-EXTERNAL>false</IS-EXTERNAL>
    <PORT-REF DEST="P-PORT-PROTOTYPE">/Components/ThrottleController/PUnlimitedThrottlePosition</PORT-REF>
    <DATA-ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE">/Interfaces/IUnlimitedThrottlePosition/UnlimitedThrottlePosition</DATA-ELEMENT-REF>
    <TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>VARIABLE-DATA-PROTOTYPE-SENT</TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>
</TD-EVENT-VARIABLE-DATA-PROTOTYPE>
<TD-EVENT-VARIABLE-DATA-PROTOTYPE>
    <SHORT-NAME>RUnlimitedThrottlePositionReceived</SHORT-NAME>
    <IS-EXTERNAL>false</IS-EXTERNAL>
    <PORT-REF DEST="R-PORT-PROTOTYPE">/Components/ThrottleActuator/RUnlimitedThrottlePosition</PORT-REF>
    <DATA-ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE">/Interfaces/IUnlimitedThrottlePosition/UnlimitedThrottlePosition</DATA-ELEMENT-REF>
    <TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>VARIABLE-DATA-PROTOTYPE-RECEIVED</TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>
</TD-EVENT-VARIABLE-DATA-PROTOTYPE>
<TD-EVENT-VARIABLE-DATA-PROTOTYPE>
    <SHORT-NAME>PUnlimitedThrottlePositionActuatorSent</SHORT-NAME>
    <IS-EXTERNAL>false</IS-EXTERNAL>
    <PORT-REF DEST="P-PORT-PROTOTYPE">/Components/ThrottleActuator/PUnlimitedThrottlePositionActuator</PORT-REF>
    <DATA-ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE">/Interfaces/IUnlimitedThrottlePosition/UnlimitedThrottlePosition</DATA-ELEMENT-REF>
    <TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>VARIABLE-DATA-PROTOTYPE-SENT</TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>
</TD-EVENT-VARIABLE-DATA-PROTOTYPE>
<TIMING-DESCRIPTION-EVENT-CHAIN>
    <SHORT-NAME>TimingChain1Seg0</SHORT-NAME>
    <STIMULUS-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/Timing/EngineControlVfbTiming1/RAcceleratorPedalPositionSensorReceived</STIMULUS-REF>
    <RESPONSE-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/Timing/EngineControlVfbTiming1/RAcceleratorPedalPositionSent</RESPONSE-REF>
    <SEGMENT-REFS>
        <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlVfbTiming1/TimingChain1Seg0</SEGMENT-REF>
    </SEGMENT-REFS>
</TIMING-DESCRIPTION-EVENT-CHAIN>
<TIMING-DESCRIPTION-EVENT-CHAIN>
    <SHORT-NAME>TimingChain1Seg1</SHORT-NAME>
    <STIMULUS-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/Timing/EngineControlVfbTiming1/RAcceleratorPedalPositionSent</STIMULUS-REF>
    <RESPONSE-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/Timing/EngineControlVfbTiming1/RAcceleratorPedalPositionReceived</RESPONSE-REF>
    <SEGMENT-REFS>
        <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlVfbTiming1/TimingChain1Seg1</SEGMENT-REF>
    </SEGMENT-REFS>
</TIMING-DESCRIPTION-EVENT-CHAIN>
<TIMING-DESCRIPTION-EVENT-CHAIN>
    <SHORT-NAME>TimingChain1Seg1_1</SHORT-NAME>
    <STIMULUS-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/Timing/EngineControlVfbTiming1/RAcceleratorPedalPositionReceived</STIMULUS-REF>
    <RESPONSE-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/Timing/EngineControlVfbTiming1/PVotedPedalPositionSent</RESPONSE-REF>
    <SEGMENT-REFS>
        <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlVfbTiming1/TimingChain1Seg1_1</SEGMENT-REF>
    </SEGMENT-REFS>
</TIMING-DESCRIPTION-EVENT-CHAIN>
<TIMING-DESCRIPTION-EVENT-CHAIN>
    <SHORT-NAME>TimingChain1Seg2</SHORT-NAME>
    <STIMULUS-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/Timing/EngineControlVfbTiming1/PVotedPedalPositionSent</STIMULUS-REF>
    <RESPONSE-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/Timing/EngineControlVfbTiming1/RVotedPedalPositionReceived</RESPONSE-REF>

```

```

<SEGMENT-REFS>
  <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlVfbTiming1/TimingChain1Seg2</SEGMENT-REF>
  </SEGMENT-REFS>
</TIMING-DESCRIPTION-EVENT-CHAIN>
<TIMING-DESCRIPTION-EVENT-CHAIN>
  <SHORT-NAME>TimingChain1Seg2_1</SHORT-NAME>
  <STIMULUS-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/Timing/EngineControlVfbTiming1/
    RVotedPedalPositionReceived</STIMULUS-REF>
  <RESPONSE-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/Timing/EngineControlVfbTiming1/
    PUnlimitedThrottlePositionSent</RESPONSE-REF>
  <SEGMENT-REFS>
    <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlVfbTiming1/TimingChain1Seg2_1</SEGMENT-REF>
    </SEGMENT-REFS>
  </TIMING-DESCRIPTION-EVENT-CHAIN>
<TIMING-DESCRIPTION-EVENT-CHAIN>
  <SHORT-NAME>TimingChain1Seg3</SHORT-NAME>
  <STIMULUS-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/Timing/EngineControlVfbTiming1/
    PUnlimitedThrottlePositionSent</STIMULUS-REF>
  <RESPONSE-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/Timing/EngineControlVfbTiming1/
    PUnlimitedThrottlePositionReceived</RESPONSE-REF>
  <SEGMENT-REFS>
    <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlVfbTiming1/TimingChain1Seg3</SEGMENT-REF>
    </SEGMENT-REFS>
  </TIMING-DESCRIPTION-EVENT-CHAIN>
<TIMING-DESCRIPTION-EVENT-CHAIN>
  <SHORT-NAME>TimingChain1Seg4</SHORT-NAME>
  <STIMULUS-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/Timing/EngineControlVfbTiming1/
    PUnlimitedThrottlePositionReceived</STIMULUS-REF>
  <RESPONSE-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/Timing/EngineControlVfbTiming1/
    PUnlimitedThrottlePositionActuatorSent</RESPONSE-REF>
  <SEGMENT-REFS>
    <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlVfbTiming1/TimingChain1Seg4</SEGMENT-REF>
    </SEGMENT-REFS>
  </TIMING-DESCRIPTION-EVENT-CHAIN>
<TIMING-DESCRIPTION-EVENT-CHAIN>
  <SHORT-NAME>TimingChain1AllSeg</SHORT-NAME>
  <STIMULUS-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/Timing/EngineControlVfbTiming1/
    RAcceleratorPedalPositionSensorReceived</STIMULUS-REF>
  <RESPONSE-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/Timing/EngineControlVfbTiming1/
    PUnlimitedThrottlePositionActuatorSent</RESPONSE-REF>
  <SEGMENT-REFS>
    <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlVfbTiming1/TimingChain1Seg0</SEGMENT-REF>
    <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlVfbTiming1/TimingChain1Seg1</SEGMENT-REF>
    <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlVfbTiming1/TimingChain1Seg1_1</SEGMENT-REF>
    <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlVfbTiming1/TimingChain1Seg2</SEGMENT-REF>
    <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlVfbTiming1/TimingChain1Seg2_1</SEGMENT-REF>
    <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlVfbTiming1/TimingChain1Seg3</SEGMENT-REF>
    <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlVfbTiming1/TimingChain1Seg4</SEGMENT-REF>
    </SEGMENT-REFS>
  </TIMING-DESCRIPTION-EVENT-CHAIN>
</TIMING-DESCRIPTIONS>
<TIMING-REQUIREMENTS>
  <LATENCY-TIMING-CONSTRAINT>
    <SHORT-NAME>TimingChain1AllSegLatency</SHORT-NAME>
    <LATENCY-CONSTRAINT-TYPE>REACTION</LATENCY-CONSTRAINT-TYPE>
    <SCOPE-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlVfbTiming1/TimingChain1AllSeg</SCOPE-REF>
    <MINIMUM>
      <CSE-CODE>1</CSE-CODE>
      <CSE-CODE-FACTOR>28</CSE-CODE-FACTOR>
    </MINIMUM>
    <MAXIMUM>
      <CSE-CODE>1</CSE-CODE>
      <CSE-CODE-FACTOR>30</CSE-CODE-FACTOR>
    </MAXIMUM>
  </LATENCY-TIMING-CONSTRAINT>
</TIMING-REQUIREMENTS>
<COMPONENT-REF DEST="COMPOSITION-SW-COMPONENT-TYPE">/Components/EngineControl</COMPONENT-REF>
</VFB-TIMING>
  
```

Listing C.22: Event Definitions and Constraints for Requirement 1

C.7.2 Application Notes - Engine Control - Requirement 2

```

<VFB-TIMING>
  <SHORT-NAME>EngineControlVfbTiming2</SHORT-NAME>
  <TIMING-DESCRIPTIONS>
    <TD-EVENT-VARIABLE-DATA-PROTOTYPE>
      <SHORT-NAME>RThrottlePositionSensorReceived</SHORT-NAME>
      <IS-EXTERNAL>false</IS-EXTERNAL>
      <PORT-REF DEST="R-PORT-PROTOTYPE">/Components/ThrottleSensor/RThrottlePositionSensor</PORT-REF>
    </TD-EVENT-VARIABLE-DATA-PROTOTYPE>
  </TIMING-DESCRIPTIONS>
</VFB-TIMING>
  
```

```

<DATA-ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE">/Interfaces/IThrottlePositionSensor/ThrottlePositionSensor</
    DATA-ELEMENT-REF>
    <TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>VARIABLE-DATA-PROTOTYPE-RECEIVED</TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>
</TD-EVENT-VARIABLE-DATA-PROTOTYPE>
<TD-EVENT-VARIABLE-DATA-PROTOTYPE>
    <SHORT-NAME>PThrottlePositionSent</SHORT-NAME>
    <IS-EXTERNAL>false</IS-EXTERNAL>
    <PORT-REF DEST="P-PORT-PROTOTYPE">/Components/ThrottleSensor/PThrottlePosition</PORT-REF>
    <DATA-ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE">/Interfaces/IThrottlePosition/ThrottlePosition</DATA-ELEMENT-
        REF>
    <TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>VARIABLE-DATA-PROTOTYPE-SENT</TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>
</TD-EVENT-VARIABLE-DATA-PROTOTYPE>
<TD-EVENT-VARIABLE-DATA-PROTOTYPE>
    <SHORT-NAME>RThrottlePositionReceived</SHORT-NAME>
    <IS-EXTERNAL>false</IS-EXTERNAL>
    <PORT-REF DEST="R-PORT-PROTOTYPE">/Components/BaseFuelMass/RThrottlePosition</PORT-REF>
    <DATA-ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE">/Interfaces/IThrottlePosition/ThrottlePosition</DATA-ELEMENT-
        REF>
    <TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>VARIABLE-DATA-PROTOTYPE-RECEIVED</TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>
</TD-EVENT-VARIABLE-DATA-PROTOTYPE>
<TIMING-DESCRIPTION-EVENT-CHAIN>
    <SHORT-NAME>TimingChain2Seg0</SHORT-NAME>
    <STIMULUS-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/Timing/EngineControlVfbTiming2/
        RThrottlePositionSensorReceived</STIMULUS-REF>
    <RESPONSE-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/Timing/EngineControlVfbTiming2/PThrottlePositionSent</
        RESPONSE-REF>
    <SEGMENT-REFS>
        <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlVfbTiming2/TimingChain2Seg0</SEGMENT-
            REF>
    </SEGMENT-REFS>
</TIMING-DESCRIPTION-EVENT-CHAIN>
<TIMING-DESCRIPTION-EVENT-CHAIN>
    <SHORT-NAME>TimingChain2Seg1</SHORT-NAME>
    <STIMULUS-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/Timing/EngineControlVfbTiming2/PThrottlePositionSent</
        STIMULUS-REF>
    <RESPONSE-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/Timing/EngineControlVfbTiming2/RThrottlePositionReceived</
        RESPONSE-REF>
    <SEGMENT-REFS>
        <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlVfbTiming2/TimingChain2Seg1</SEGMENT-
            REF>
    </SEGMENT-REFS>
</TIMING-DESCRIPTION-EVENT-CHAIN>
<TIMING-DESCRIPTION-EVENT-CHAIN>
    <SHORT-NAME>TimingChain2AllSeg</SHORT-NAME>
    <STIMULUS-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/Timing/EngineControlVfbTiming2/
        RThrottlePositionSensorReceived</STIMULUS-REF>
    <RESPONSE-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/Timing/EngineControlVfbTiming2/RThrottlePositionReceived</
        RESPONSE-REF>
    <SEGMENT-REFS>
        <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlVfbTiming2/TimingChain2Seg0</SEGMENT-
            REF>
        <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlVfbTiming2/TimingChain2Seg1</SEGMENT-
            REF>
    </SEGMENT-REFS>
</TIMING-DESCRIPTION-EVENT-CHAIN>
<TIMING-DESCRIPTIONS>
<TIMING-REQUIREMENTS>
    <LATENCY-TIMING-CONSTRAINT>
        <SHORT-NAME>TimingChain2AllSegLatency</SHORT-NAME>
        <LATENCY-CONSTRAINT-TYPE>AGE</LATENCY-CONSTRAINT-TYPE>
        <SCOPE-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlVfbTiming2/TimingChain2Seg1</SCOPE-REF>
        <MAXIMUM>
            <CSE-CODE>3</CSE-CODE>
            <CSE-CODE-FACTOR>10</CSE-CODE-FACTOR>
        </MAXIMUM>
        <LATENCY-TIMING-CONSTRAINT>
    </TIMING-REQUIREMENTS>
</VFB-TIMING>
    
```

Listing C.23: Event Definitions and Constraints for Requirement 2

C.7.3 Application Notes - Engine Control - Requirement 3

```

<VFB-TIMING>
    <SHORT-NAME>EngineControlVfbTiming3</SHORT-NAME>
    <TIMING-DESCRIPTIONS>
        <TD-EVENT-VARIABLE-DATA-PROTOTYPE>
            <SHORT-NAME>PMafRateOutSent</SHORT-NAME>
            <IS-EXTERNAL>false</IS-EXTERNAL>
            <PORT-REF DEST="P-PORT-PROTOTYPE">/Components/BaseFuelMass/PMafRateOut</PORT-REF>
            <DATA-ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE">/Interfaces/IMafRateOut/MafRateOut</DATA-ELEMENT-REF>
            <TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>VARIABLE-DATA-PROTOTYPE-SENT</TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>
        </TD-EVENT-VARIABLE-DATA-PROTOTYPE>
        <TD-EVENT-VARIABLE-DATA-PROTOTYPE>
            <SHORT-NAME>RMafRateOutReceived</SHORT-NAME>
            <IS-EXTERNAL>false</IS-EXTERNAL>
            <PORT-REF DEST="R-PORT-PROTOTYPE">/Components/Ignition/RMafRateOut</PORT-REF>
            <DATA-ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE">/Interfaces/IMafRateOut/MafRateOut</DATA-ELEMENT-REF>
            <TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>VARIABLE-DATA-PROTOTYPE-RECEIVED</TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>
        </TD-EVENT-VARIABLE-DATA-PROTOTYPE>
        <TD-EVENT-VARIABLE-DATA-PROTOTYPE>
    
```

```

<SHORT-NAME>PIgnitionTimingSent</SHORT-NAME>
<IS-EXTERNAL>false</IS-EXTERNAL>
<PORT-REF DEST="P-PORT-PROTOTYPE">/Components/Ignition/PIgnitionTiming</PORT-REF>
<DATA-ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE">/Interfaces/IIgnitionTiming/IgnitionTiming</DATA-ELEMENT-REF>
<TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>VARIABLE-DATA-PROTOTYPE-SENT</TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>
</TD-EVENT-VARIABLE-DATA-PROTOTYPE>
<TD-EVENT-VARIABLE-DATA-PROTOTYPE>
<SHORT-NAME>RIgnitionTimingReceived</SHORT-NAME>
<IS-EXTERNAL>false</IS-EXTERNAL>
<PORT-REF DEST="R-PORT-PROTOTYPE">/Components/IgnitionActuator/RIgnitionTiming</PORT-REF>
<DATA-ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE">/Interfaces/IIgnitionTiming/IgnitionTiming</DATA-ELEMENT-REF>
<TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>VARIABLE-DATA-PROTOTYPE-RECEIVED</TD-EVENT-VARIABLE-DATA-PROTOTYPE-TYPE>
</TD-EVENT-VARIABLE-DATA-PROTOTYPE>
<TIMING-DESCRIPTION-EVENT-CHAIN>
<SHORT-NAME>TimingChain3Seg0</SHORT-NAME>
<STIMULUS-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/Timing/EngineControlVfbTiming1/
    RUUnlimitedThrottlePositionReceived</STIMULUS-REF>
<RESPONSE-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/Timing/EngineControlVfbTiming3/PMafRateOutSent</RESPONSE-REF>
<SEGMENT-REFS>
    <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlVfbTiming3/TimingChain3Seg0</SEGMENT-REF>
</SEGMENT-REFS>
</TIMING-DESCRIPTION-EVENT-CHAIN>
<TIMING-DESCRIPTION-EVENT-CHAIN>
<SHORT-NAME>TimingChain3Seg1</SHORT-NAME>
<STIMULUS-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/Timing/EngineControlVfbTiming3/PMafRateOutSent</STIMULUS-REF>
<RESPONSE-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/Timing/EngineControlVfbTiming3/RMafRateOutReceived</RESPONSE-REF>
<SEGMENT-REFS>
    <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlVfbTiming3/TimingChain3Seg1_1</SEGMENT-REF>
</SEGMENT-REFS>
</TIMING-DESCRIPTION-EVENT-CHAIN>
<TIMING-DESCRIPTION-EVENT-CHAIN>
<SHORT-NAME>TimingChain3Seg1_1</SHORT-NAME>
<STIMULUS-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/Timing/EngineControlVfbTiming3/RMafRateOutReceived</STIMULUS-REF>
<RESPONSE-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/Timing/EngineControlVfbTiming3/PIgnitionTimingSent</RESPONSE-REF>
<SEGMENT-REFS>
    <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlVfbTiming3/TimingChain3Seg1_1</SEGMENT-REF>
</SEGMENT-REFS>
</TIMING-DESCRIPTION-EVENT-CHAIN>
<TIMING-DESCRIPTION-EVENT-CHAIN>
<SHORT-NAME>TimingChain3Seg2</SHORT-NAME>
<STIMULUS-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/Timing/EngineControlVfbTiming3/PIgnitionTimingSent</STIMULUS-REF>
<RESPONSE-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/Timing/EngineControlVfbTiming3/RIgnitionTimingReceived</RESPONSE-REF>
<SEGMENT-REFS>
    <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlVfbTiming3/TimingChain3Seg2</SEGMENT-REF>
</SEGMENT-REFS>
</TIMING-DESCRIPTION-EVENT-CHAIN>
<TIMING-DESCRIPTION-EVENT-CHAIN>
<SHORT-NAME>TimingChain3AllSeg</SHORT-NAME>
<STIMULUS-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/Timing/EngineControlVfbTiming1/
    AcceleratorPedalPositionSensorReceived</STIMULUS-REF>
<RESPONSE-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/Timing/EngineControlVfbTiming3/RIgnitionTimingReceived</RESPONSE-REF>
<SEGMENT-REFS>
    <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlVfbTiming1/TimingChain1AllSeg</SEGMENT-REF>
    <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlVfbTiming2/TimingChain2Seg0</SEGMENT-REF>
    <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlVfbTiming2/TimingChain2Seg1</SEGMENT-REF>
    <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlVfbTiming3/TimingChain3Seg0</SEGMENT-REF>
    <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlVfbTiming3/TimingChain3Seg1</SEGMENT-REF>
    <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlVfbTiming3/TimingChain3Seg1_1</SEGMENT-REF>
    <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlVfbTiming3/TimingChain3Seg2</SEGMENT-REF>
</SEGMENT-REFS>
</TIMING-DESCRIPTION-EVENT-CHAIN>
</TIMING-DESCRIPTIONS>
<TIMING-GUARANTEES>
    <LATENCY-TIMING-CONSTRAINT>
        <SHORT-NAME>TimingChain3AllSegLatency1</SHORT-NAME>
        <LATENCY-CONSTRAINT-TYPE>REACTION</LATENCY-CONSTRAINT-TYPE>
        <SCOPE-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlVfbTiming3/TimingChain3AllSeg</SCOPE-REF>
        <MAXIMUM>
            <CSE-CODE>3</CSE-CODE>
            <CSE-CODE-FACTOR>50</CSE-CODE-FACTOR>
        </MAXIMUM>
    </LATENCY-TIMING-CONSTRAINT>
</TIMING-GUARANTEES>
<TIMING-REQUIREMENTS>
    <OFFSET-TIMING-CONSTRAINT>
        <SHORT-NAME>Otc</SHORT-NAME>

```

```

<SOURCE-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/Timing/EngineControlVfbTiming1/
  PUnlimitedThrottlePositionActuatorSent</SOURCE-REF>
<TARGET-REF DEST="TD-EVENT-VARIABLE-DATA-PROTOTYPE">/Timing/EngineControlVfbTiming2/RThrottlePositionReceived</
  TARGET-REF>
<MAXIMUM>
  <CSE-CODE>3</CSE-CODE>
  <CSE-CODE-FACTOR>5</CSE-CODE-FACTOR>
</MAXIMUM>
</OFFSET-TIMING-CONSTRAINT>
<LATENCY-TIMING-CONSTRAINT>
  <SHORT-NAME>TimingChain3AllSegLatency2</SHORT-NAME>
  <LATENCY-CONSTRAINT-TYPE>REACTION</LATENCY-CONSTRAINT-TYPE>
  <SCOPE-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlVfbTiming3/TimingChain3AllSeg</SCOPE-REF>
<MAXIMUM>
  <CSE-CODE>3</CSE-CODE>
  <CSE-CODE-FACTOR>50</CSE-CODE-FACTOR>
</MAXIMUM>
</LATENCY-TIMING-CONSTRAINT>
</TIMING-REQUIREMENTS>
</VFB-TIMING>
  
```

Listing C.24: Event Definitions and Constraints for Requirement 3

C.7.4 Application Notes - Engine Control - Requirement 4

```

<ECU-TIMING>
  <SHORT-NAME>EngineControlEcuTiming</SHORT-NAME>
  <TIMING-DESCRIPTIONS>
    <TD-EVENT-BSW-INTERNAL-BEHAVIOR>
      <SHORT-NAME>CamShaftISRActivated</SHORT-NAME>
      <BSW-MODULE-ENTITY-REF DEST="BSW-INTERRUPT-ENTITY">/Modules/CamShaft/CamShaftBehavior/CamShaftISR</BSW-MODULE-
        ENTITY-REF>
      <TD-EVENT-BSW-INTERNAL-BEHAVIOR-TYPE>BSW-MODULE-ENTITY-ACTIVATED</TD-EVENT-BSW-INTERNAL-BEHAVIOR-TYPE>
    </TD-EVENT-BSW-INTERNAL-BEHAVIOR>
    <TD-EVENT-BSW-INTERNAL-BEHAVIOR>
      <SHORT-NAME>CamShaftISRStarted</SHORT-NAME>
      <BSW-MODULE-ENTITY-REF DEST="BSW-INTERRUPT-ENTITY">/Modules/CamShaft/CamShaftBehavior/CamShaftISR</BSW-MODULE-
        ENTITY-REF>
      <TD-EVENT-BSW-INTERNAL-BEHAVIOR-TYPE>BSW-MODULE-ENTITY-STARTED</TD-EVENT-BSW-INTERNAL-BEHAVIOR-TYPE>
    </TD-EVENT-BSW-INTERNAL-BEHAVIOR>
    <TD-EVENT-BSW-INTERNAL-BEHAVIOR>
      <SHORT-NAME>CamShaftIsrTerminated</SHORT-NAME>
      <BSW-MODULE-ENTITY-REF DEST="BSW-INTERRUPT-ENTITY">/Modules/CamShaft/CamShaftBehavior/CamShaftISR</BSW-MODULE-
        ENTITY-REF>
      <TD-EVENT-BSW-INTERNAL-BEHAVIOR-TYPE>BSW-MODULE-ENTITY-TERMINATED</TD-EVENT-BSW-INTERNAL-BEHAVIOR-TYPE>
    </TD-EVENT-BSW-INTERNAL-BEHAVIOR>
    <TD-EVENT-SWC-INTERNAL-BEHAVIOR>
      <SHORT-NAME>IgnitionActuatorCalculationActivated</SHORT-NAME>
      <RUNNABLE-REF DEST="RUNNABLE-ENTITY">/Components/IngnitionActuator/IgnitionActuatorBehavior/
        IgnitionActuatorCalculation</RUNNABLE-REF>
      <TD-EVENT-SWC-INTERNAL-BEHAVIOR-TYPE>RUNNABLE-ENTITY-ACTIVATED</TD-EVENT-SWC-INTERNAL-BEHAVIOR-TYPE>
    </TD-EVENT-SWC-INTERNAL-BEHAVIOR>
    <TD-EVENT-SWC-INTERNAL-BEHAVIOR>
      <SHORT-NAME>IgnitionActuatorCalculationStarted</SHORT-NAME>
      <RUNNABLE-REF DEST="RUNNABLE-ENTITY">/Components/IngnitionActuator/IgnitionActuatorBehavior/
        IgnitionActuatorCalculation</RUNNABLE-REF>
      <TD-EVENT-SWC-INTERNAL-BEHAVIOR-TYPE>RUNNABLE-ENTITY-STARTED</TD-EVENT-SWC-INTERNAL-BEHAVIOR-TYPE>
    </TD-EVENT-SWC-INTERNAL-BEHAVIOR>
    <TD-EVENT-SWC-INTERNAL-BEHAVIOR>
      <SHORT-NAME>IgnitionActuatorCalculationTerminated</SHORT-NAME>
      <RUNNABLE-REF DEST="RUNNABLE-ENTITY">/Components/IngnitionActuator/IgnitionActuatorBehavior/
        IgnitionActuatorCalculation</RUNNABLE-REF>
      <TD-EVENT-SWC-INTERNAL-BEHAVIOR-TYPE>RUNNABLE-ENTITY-TERMINATED</TD-EVENT-SWC-INTERNAL-BEHAVIOR-TYPE>
    </TD-EVENT-SWC-INTERNAL-BEHAVIOR>
    <TIMING-DESCRIPTION-EVENT-CHAIN>
      <SHORT-NAME>TimingChain4Seg1</SHORT-NAME>
      <STIMULUS-REF DEST="TD-EVENT-BSW-INTERNAL-BEHAVIOR">/Timing/EngineControlEcuTiming/CamShaftISRStarted</STIMULUS-
        REF>
      <RESPONSE-REF DEST="TD-EVENT-BSW-INTERNAL-BEHAVIOR">/Timing/EngineControlEcuTiming/CamShaftIsrTerminated</
        RESPONSE-REF>
      <SEGMENT-REFS>
        <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlEcuTiming/TimingChain4Seg1</SEGMENT-
          REF>
      </SEGMENT-REFS>
    </TIMING-DESCRIPTION-EVENT-CHAIN>
    <TIMING-DESCRIPTION-EVENT-CHAIN>
      <SHORT-NAME>TimingChain4Seg2</SHORT-NAME>
      <STIMULUS-REF DEST="TD-EVENT-BSW-INTERNAL-BEHAVIOR">/Timing/EngineControlEcuTiming/CamShaftIsrTerminated</
        STIMULUS-REF>
      <RESPONSE-REF DEST="TD-EVENT-SWC-INTERNAL-BEHAVIOR">/Timing/EngineControlEcuTiming/
        IgnitionActuatorCalculationStarted</RESPONSE-REF>
      <SEGMENT-REFS>
        <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlEcuTiming/TimingChain4Seg2</SEGMENT-
          REF>
      </SEGMENT-REFS>
    </TIMING-DESCRIPTION-EVENT-CHAIN>
    <TIMING-DESCRIPTION-EVENT-CHAIN>
      <SHORT-NAME>TimingChain4Seg3</SHORT-NAME>
      <STIMULUS-REF DEST="TD-EVENT-SWC-INTERNAL-BEHAVIOR">/Timing/EngineControlEcuTiming/
        IgnitionActuatorCalculationStarted</STIMULUS-REF>
      <RESPONSE-REF DEST="TD-EVENT-SWC-INTERNAL-BEHAVIOR">/Timing/EngineControlEcuTiming/
        IgnitionActuatorCalculationTerminated</RESPONSE-REF>
    </TIMING-DESCRIPTION-EVENT-CHAIN>
  </TIMING-DESCRIPTIONS>
  
```

```

<SEGMENT-REFS>
  <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlEcuTiming/TimingChain4Seg3</SEGMENT-
  REF>
</SEGMENT-REFS>
</TIMING-DESCRIPTION-EVENT-CHAIN>
<TIMING-DESCRIPTION-EVENT-CHAIN>
  <SHORT-NAME>TimingChain4AllSeg</SHORT-NAME>
  <STIMULUS-REF DEST="TD-EVENT-BSW-INTERNAL-BEHAVIOR">/Timing/EngineControlEcuTiming/CamShaftISRStarted</STIMULUS-
  REF>
  <RESPONSE-REF DEST="TD-EVENT-SWC-INTERNAL-BEHAVIOR">/Timing/EngineControlEcuTiming/
    IgnitionActuatorCalculationTerminated</RESPONSE-REF>
<SEGMENT-REFS>
  <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlEcuTiming/TimingChain4Seg1</SEGMENT-
  REF>
  <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlEcuTiming/TimingChain4Seg2</SEGMENT-
  REF>
  <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlEcuTiming/TimingChain4Seg3</SEGMENT-
  REF>
</SEGMENT-REFS>
</TIMING-DESCRIPTION-EVENT-CHAIN>
</TIMING-DESCRIPTIONS>
<TIMING-REQUIREMENTS>
  <LATENCY-TIMING-CONSTRAINT>
    <SHORT-NAME>TimingChain4AllSegLatency</SHORT-NAME>
    <LATENCY-CONSTRAINT-TYPE>REACTION</LATENCY-CONSTRAINT-TYPE>
    <SCOPE-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlEcuTiming/TimingChain4AllSeg</SCOPE-REF>
    <MAXIMUM>
      <CSE-CODE>3</CSE-CODE>
      <CSE-CODE-FACTOR>3</CSE-CODE-FACTOR>
    </MAXIMUM>
  </LATENCY-TIMING-CONSTRAINT>
</TIMING-REQUIREMENTS>
</ECU-TIMING>
  
```

Listing C.25: Event Definitions and Constraints for Requirement 4

C.7.5 Application Notes - Engine Control - Requirement 5

```

<SWC-TIMING>
  <SHORT-NAME>EngineControlSwcTimingIgnition</SHORT-NAME>
  <TIMING-DESCRIPTIONS>
    <TD-EVENT-SWC-INTERNAL-BEHAVIOR>
      <SHORT-NAME>IgnitionTimingCalculationActivated</SHORT-NAME>
      <RUNNABLE-REF DEST="RUNNABLE-ENTITY">/Components/Ignition/IgnitionBehavior/IgnitionTimingCalculation</RUNNABLE-
      REF>
      <TD-EVENT-SWC-INTERNAL-BEHAVIOR-TYPE>RUNNABLE-ENTITY-ACTIVATED</TD-EVENT-SWC-INTERNAL-BEHAVIOR-TYPE>
    </TD-EVENT-SWC-INTERNAL-BEHAVIOR>
    <TD-EVENT-SWC-INTERNAL-BEHAVIOR>
      <SHORT-NAME>IgnitionTimingCalculationStarted</SHORT-NAME>
      <RUNNABLE-REF DEST="RUNNABLE-ENTITY">/Components/Ignition/IgnitionBehavior/IgnitionTimingCalculation</RUNNABLE-
      REF>
      <TD-EVENT-SWC-INTERNAL-BEHAVIOR-TYPE>RUNNABLE-ENTITY-STARTED</TD-EVENT-SWC-INTERNAL-BEHAVIOR-TYPE>
    </TD-EVENT-SWC-INTERNAL-BEHAVIOR>
    <TD-EVENT-SWC-INTERNAL-BEHAVIOR>
      <SHORT-NAME>IgnitionTimingCalculationTerminated</SHORT-NAME>
      <RUNNABLE-REF DEST="RUNNABLE-ENTITY">/Components/Ignition/IgnitionBehavior/IgnitionTimingCalculation</RUNNABLE-
      REF>
      <TD-EVENT-SWC-INTERNAL-BEHAVIOR-TYPE>RUNNABLE-ENTITY-TERMINATED</TD-EVENT-SWC-INTERNAL-BEHAVIOR-TYPE>
    </TD-EVENT-SWC-INTERNAL-BEHAVIOR>
    <TIMING-DESCRIPTION-EVENT-CHAIN>
      <SHORT-NAME>TimingChain5</SHORT-NAME>
      <STIMULUS-REF DEST="TD-EVENT-SWC-INTERNAL-BEHAVIOR">/Timing/EngineControlSwcTimingIgnition/
        IgnitionTimingCalculationActivated</STIMULUS-REF>
      <RESPONSE-REF DEST="TD-EVENT-SWC-INTERNAL-BEHAVIOR">/Timing/EngineControlSwcTimingIgnition/
        IgnitionTimingCalculationTerminated</RESPONSE-REF>
      <SEGMENT-REFS>
        <SEGMENT-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlSwcTimingIgnition/TimingChain5</
        SEGMENT-REF>
      </SEGMENT-REFS>
    </TIMING-DESCRIPTION-EVENT-CHAIN>
  </TIMING-DESCRIPTIONS>
  <TIMING-REQUIREMENTS>
    <PERIODIC-EVENT-TRIGGERING>
      <SHORT-NAME>CalculateIgnitionTimingTriggeringConstraint</SHORT-NAME>
      <JITTER>
        <CSE-CODE>3</CSE-CODE>
        <CSE-CODE-FACTOR>1</CSE-CODE-FACTOR>
      </JITTER>
      <PERIOD>
        <CSE-CODE>3</CSE-CODE>
        <CSE-CODE-FACTOR>20</CSE-CODE-FACTOR>
      </PERIOD>
    </PERIODIC-EVENT-TRIGGERING>
    <LATENCY-TIMING-CONSTRAINT>
      <SHORT-NAME>TimingChain5Latency</SHORT-NAME>
      <LATENCY-CONSTRAINT-TYPE>REACTION</LATENCY-CONSTRAINT-TYPE>
      <SCOPE-REF DEST="TIMING-DESCRIPTION-EVENT-CHAIN">/Timing/EngineControlSwcTimingIgnition/TimingChain5</SCOPE-REF>
      <MAXIMUM>
        <CSE-CODE>3</CSE-CODE>
        <CSE-CODE-FACTOR>20</CSE-CODE-FACTOR>
      </MAXIMUM>
    </LATENCY-TIMING-CONSTRAINT>
  </TIMING-REQUIREMENTS>
  
```

```
</LATENCY-TIMING-CONSTRAINT>
</TIMING-REQUIREMENTS>
<BEHAVIOR-REF DEST="SWC-INTERNAL-BEHAVIOR">/Components/Ignition/IgnitionBehavior</BEHAVIOR-REF>
</SWC-TIMING>
</ELEMENTS>
```

Listing C.26: Event Definitions and Constraints for Requirement 5

D Mentioned Class Tables

For the sake of completeness, this chapter contains a set of class tables representing meta-classes mentioned in the context of this document but which are not contained directly in the scope of describing specific meta-model semantics.

| Class | ARPackage | | | |
|---------------|--|-------|------|--|
| Package | M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::ARPackage | | | |
| Note | AUTOSAR package, allowing to create top level packages to structure the contained ARElements. ARPackages are open sets. This means that in a file based description system multiple files can be used to partially describe the contents of a package. This is an extended version of MSR's SW-SYSTEM. | | | |
| Base | <i>ARObject</i> , <i>AtpBlueprint</i> , <i>AtpBlueprintable</i> , <i>CollectableElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i> | | | |
| Aggregated by | <i>ARPackage.arPackage</i> , <i>AUTOSAR.arPackage</i> | | | |
| Attribute | Type | Mult. | Kind | Note |
| arPackage | <i>ARPackage</i> | * | aggr | This represents a sub package within an ARPackage, thus allowing for an unlimited package hierarchy. Stereotypes: <i>atpSplittable</i> ; <i>atpVariation</i> Tags: atp.Splitkey=arPackage.shortName, arPackage.variation Point.shortLabel vh.latestBindingTime=blueprintDerivationTime xml.sequenceOffset=30 |
| element | <i>PackageableElement</i> | * | aggr | Elements that are part of this package Stereotypes: <i>atpSplittable</i> ; <i>atpVariation</i> Tags: atp.Splitkey=element.shortName, element.variation Point.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=20 |
| referenceBase | <i>ReferenceBase</i> | * | aggr | This denotes the reference bases for the package. This is the basis for all relative references within the package. The base needs to be selected according to the base attribute within the references. Stereotypes: <i>atpSplittable</i> Tags: atp.Splitkey=referenceBase.shortLabel xml.sequenceOffset=10 |

Table D.1: ARPackage

| Class | AUTOSAR | | | |
|-----------|---|-------|------|------|
| Package | M2::AUTOSARTemplates::AutosarTopLevelStructure | | | |
| Note | Root element of an AUTOSAR description, also the root element in corresponding XML documents. Tags: <i>xml.globalElement=true</i> | | | |
| Base | <i>ARObject</i> | | | |
| Attribute | Type | Mult. | Kind | Note |





| Class | AUTOSAR | | | |
|------------------|--------------------|------|------|---|
| adminData | AdminData | 0..1 | aggr | <p>This represents the administrative data of an Autosar file.</p> <p>Stereotypes: atpSplittable</p> <p>Tags:</p> <p>atp.Splitkey=adminData xml.sequenceOffset=10</p> |
| arPackage | ARPackage | * | aggr | <p>This is the top level package in an AUTOSAR model.</p> <p>Stereotypes: atpSplittable; atpVariation</p> <p>Tags:</p> <p>atp.Splitkey=arPackage.shortName, arPackage.variation Point.shortLabel vh.latestBindingTime=blueprintDerivationTime xml.sequenceOffset=30</p> |
| fileInfo Comment | FileInfoComment | 0..1 | aggr | <p>This represents a possibility to provide a structured comment in an AUTOSAR file.</p> <p>Stereotypes: atpStructuredComment</p> <p>Tags:</p> <p>xml.roleElement=true xml.sequenceOffset=-10 xml.typeElement=false</p> |
| introduction | DocumentationBlock | 0..1 | aggr | <p>This represents an introduction on the Autosar file. It is intended for example to represent disclaimers and legal notes.</p> <p>Tags: xml.sequenceOffset=20</p> |

Table D.2: AUTOSAR

| Class | <i>AbstractEvent</i> (abstract) | | | |
|----------------------------------|---|-------|------|--|
| Package | M2::AUTOSARTemplates::CommonStructure::InternalBehavior | | | |
| Note | This meta-class represents the abstract ability to model an event that can be taken to implement application software or basic software in AUTOSAR. | | | |
| Base | ARObject, <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i> | | | |
| Subclasses | <i>BswEvent</i> , <i>RTEEvent</i> | | | |
| Attribute | Type | Mult. | Kind | Note |
| activation Reason Representation | ExecutableEntity ActivationReason | 0..1 | ref | If the activationReasonRepresentation is referenced from the enclosing AbstractEvent this shall be taken as an indication that the latter contributes to the activating vector of this ExecutableEntity that owns the referenced ExecutableEntityActivationReason. |

Table D.3: AbstractEvent

| Class | <i>AbstractRequiredPortPrototype</i> (abstract) | | | |
|---------------|--|-------|------|------|
| Package | M2::AUTOSARTemplates::SWComponentTemplate::Components | | | |
| Note | This abstract class provides the ability to become a required PortPrototype. | | | |
| Base | ARObject, <i>AtpBlueprintable</i> , <i>AtpFeature</i> , <i>AtpPrototype</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Port Prototype</i> , <i>Referrable</i> | | | |
| Subclasses | <i>PRPortPrototype</i> , <i>RPortPrototype</i> | | | |
| Aggregated by | <i>AtpClassifier</i> . <i>atpFeature</i> , <i>SwComponentType</i> . <i>port</i> | | | |
| Attribute | Type | Mult. | Kind | Note |



△

| Class | AbstractRequiredPortPrototype (abstract) | | | |
|-----------------|---|---|------|---|
| requiredComSpec | RPortComSpec | * | aggr | Required communication attributes, one for each interface element. Stereotypes: atpSplitable Tags: atp.Splitkey=requiredComSpec |

Table D.4: AbstractRequiredPortPrototype

| Class | ApplicationSwComponentType | | | |
|---------------|---|-------|------|------|
| Package | M2::AUTOSARTemplates::SWComponentTemplate::Components | | | |
| Note | The ApplicationSwComponentType is used to represent the application software. Tags: atp.recommendedPackage=SwComponentTypes | | | |
| Base | ARElement, ARObject, AtomicSwComponentType , AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, CollectableElement, Identifiable , MultilanguageReferrable, PackageableElement, Referrable , SwComponentType | | | |
| Aggregated by | ARPackage.element | | | |
| Attribute | Type | Mult. | Kind | Note |
| – | – | – | – | – |

Table D.5: ApplicationSwComponentType

| Class | AtomicSwComponentType (abstract) | | | |
|------------------|---|-------|------|--|
| Package | M2::AUTOSARTemplates::SWComponentTemplate::Components | | | |
| Note | An atomic software component is atomic in the sense that it cannot be further decomposed and distributed across multiple ECUs. | | | |
| Base | ARElement, ARObject, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, CollectableElement, Identifiable , MultilanguageReferrable, PackageableElement, Referrable , SwComponentType | | | |
| Subclasses | ApplicationSwComponentType , ComplexDeviceDriverSwComponentType , EcuAbstractionSwComponentType , NvBlockSwComponentType , SensorActuatorSwComponentType , ServiceProxySwComponentType , ServiceSwComponentType | | | |
| Aggregated by | ARPackage.element | | | |
| Attribute | Type | Mult. | Kind | Note |
| internalBehavior | SwInternalBehavior | 0..1 | aggr | The SwInternalBehaviors owned by an AtomicSwComponentType can be located in a different physical file. Therefore the aggregation is <<atpSplitable>>. Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=internalBehavior.shortName, internalBehavior.variationPoint.shortLabel vh.latestBindingTime=preCompileTime |
| symbolProps | SymbolProps | 0..1 | aggr | This represents the SymbolProps for the AtomicSwComponentType. Stereotypes: atpSplitable Tags: atp.Splitkey=symbolProps.shortName |

Table D.6: AtomicSwComponentType

| | | | | |
|------------------------------|--|--------------|-------------|---|
| Class | AtpInstanceRef (abstract) | | | |
| Package | M2::AUTOSARTemplates::GenericStructure::AbstractStructure | | | |
| Note | <p>An M0 instance of a classifier may be represented as a tree rooted at that instance, where under each node come the sub-trees representing the instances which act as features under that node.</p> <p>An instance ref specifies a navigation path from any M0 tree-instance of the base (which is a classifier) to a leaf (which is an instance of the target).</p> | | | |
| Base | ARObject | | | |
| Subclasses | AnyInstanceRef, ApplicationCompositeElementInPortInterfaceInstanceRef, ComponentInCompositionInstanceRef, ComponentInSystemInstanceRef, DataPrototypeInPortInterfaceInstanceRef, DataPrototypelnSystemInstanceRef, InnerDataPrototypeGroupInCompositionInstanceRef, InnerPortGroupInCompositionInstanceRef, InnerRunnableEntityGroupInCompositionInstanceRef, InstanceEventInCompositionInstanceRef, ModeDeclarationGroupPrototypeInSystemInstanceRef, ModeGroupInAtomicSwcInstanceRef, ModelInBswModuleDescriptionInstanceRef, ModelInSwcInstanceRef, OperationArgumentInComponentInstanceRef, OperationInAtomicSwcInstanceRef, OperationInSystemInstanceRef, PModelInSystemInstanceRef, ParameterDataPrototypeInSystemInstanceRef, ParameterInAtomicSwctypeInstanceRef, PortGroupInSystemInstanceRef, PortInCompositionTypeInstanceRef, RModelInAtomicSwcInstanceRef, RteEventInCompositionInstanceRef, RteEventInEcuInstanceRef, RteEventInSystemInstanceRef, RunnableEntityInCompositionInstanceRef, SwcServiceDependencyInSystemInstanceRef, TriggerInAtomicSwcInstanceRef, TriggerInSystemInstanceRef, VariableAccessInEcuInstanceRef, VariableDataPrototypeInCompositionInstanceRef, VariableDataPrototypeInSystemInstanceRef, VariableInAtomicSWctypeInstanceRef, VariableInAtomicSwcInstanceRef, VariableInComponentInstanceRef | | | |
| Attribute | Type | Mult. | Kind | Note |
| atpBase | AtpClassifier | 1 | ref | <p>This is the base from which the navigation path starts.</p> <p>Stereotypes: atpAbstract; atpDerived</p> |
| atpContext Element (ordered) | AtpPrototype | * | ref | <p>This is one particular step in the navigation path.</p> <p>Stereotypes: atpAbstract</p> |
| atpTarget | AtpFeature | 1 | ref | <p>This is the target of the instance ref. In other words it is the terminal of the navigation path.</p> <p>Stereotypes: atpAbstract</p> |

Table D.7: AtpInstanceRef

| | | | | |
|------------------|---|--|--|--|
| Primitive | Boolean | | | |
| Package | M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::PrimitiveTypes | | | |
| Note | <p>A Boolean value denotes a logical condition that is either 'true' or 'false'. It can be one of "0", "1", "true", "false"</p> <p>Tags: xml.xsd.customType=BOOLEAN xml.xsd.pattern=0 1 true false xml.xsd.type=string </p> | | | |

Table D.8: Boolean

| | | | | |
|----------------------|--|--------------|-------------|-------------|
| Class | BswEvent (abstract) | | | |
| Package | M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior | | | |
| Note | <p>Base class of various kinds of events which are used to trigger a BswModuleEntity of this BSW module or cluster. The event is local to the BSW module or cluster. The short name of the meta-class instance is intended as an input to configure the required API of the BSW Scheduler.</p> | | | |
| Base | ARObject, AbstractEvent, Identifiable, MultilanguageReferrable, Referrable | | | |
| Subclasses | BswInterruptEvent, BswOperationInvokedEvent, BswScheduleEvent | | | |
| Aggregated by | BswInternalBehavior.event | | | |
| Attribute | Type | Mult. | Kind | Note |





| Class | BswEvent (abstract) | | | |
|--------------------|----------------------------|------|------|--|
| context Limitation | BswDistinguished Partition | * | ref | The existence of this reference indicates that the usage of the event is limited to the context of the referred Bsw DistinguishedPartitions. |
| disabledInMode | ModeDeclaration | * | iref | <p>The modes, in which this event is disabled.</p> <p>Stereotypes: atpSplitable</p> <p>Tags: atp.Splitkey=disabledInMode.contextMode DeclarationGroup, disabledInMode.targetMode</p> <p>InstanceRef implemented by: ModelInBswModule DescriptionInstanceRef</p> |
| startsOnEvent | BswModuleEntity | 0..1 | ref | The entity which is started by the event. |

 Table D.9: **BswEvent**

| Class | BswImplementation | | | |
|-----------------------------|---|-------|------|---|
| Package | M2::AUTOSARTemplates::BswModuleTemplate::BswImplementation | | | |
| Note | Contains the implementation specific information in addition to the generic specification (BswModule Description and BswBehavior). It is possible to have several different BswImplementations referring to the same BswBehavior. Tags: atp.recommendedPackage=BswImplementations | | | |
| Base | ARElement, ARObject, CollectableElement, Identifiable , Implementation, MultilanguageReferrable, PackageableElement, Referrable | | | |
| Aggregated by | ARPackage.element | | | |
| Attribute | Type | Mult. | Kind | Note |
| arRelease Version | RevisionLabelString | 0..1 | attr | <p>Version of the AUTOSAR Release on which this implementation is based. The numbering contains three levels (major, minor, revision) which are defined by AUTOSAR.</p> |
| behavior | BswInternalBehavior | 0..1 | ref | <p>The behavior of this implementation.</p> <p>This relation is made as an association because</p> <ul style="list-style-type: none"> • it follows the pattern of the SWCT • since ARElement cannot be split, but we want supply the implementation later, the BswImplementation is not aggregated in BswBehavior |
| preconfigured Configuration | EcucModule ConfigurationValues | * | ref | <p>Reference to the set of preconfigured (i.e. fixed) configuration values for this BswImplementation.</p> <p>If the BswImplementation represents a cluster of several modules, more than one EcucModuleConfigurationValues element can be referred (at most one per module), otherwise at most one such element can be referred.</p> <p>Tags: xml.roleWrapperElement=true</p> |
| recommended Configuration | EcucModule ConfigurationValues | * | ref | Reference to one or more sets of recommended configuration values for this module or module cluster. |





| Class | BswImplementation | | | |
|-------------------------|-------------------|------|------|---|
| vendorApiInfix | Identifier | 0..1 | attr | <p>In driver modules which can be instantiated several times on a single ECU, SRS_BSW_00347 requires that the names of files, APIs, published parameters and memory allocation keywords are extended by the vendorId and a vendor specific name. This parameter is used to specify the vendor specific name. In total, the implementation specific API name is generated as follows: <Module Name>_<vendorId>_<vendorApiInfix>_<API name from SWS>.</p> <p>E.g. assuming that the vendorId of the implementer is 123 and the implementer chose a vendorApiInfix of "v11r456" an API name Can_Write defined in the SWS will translate to Can_123_v11r456_Write.</p> <p>This attribute is mandatory for all modules with upper multiplicity > 1. It shall not be used for modules with upper multiplicity =1.</p> <p>See also SWS_BSW_00102.</p> |
| vendorSpecificModuleDef | EcucModuleDef | * | ref | <p>Reference to</p> <ul style="list-style-type: none"> the vendor specific EcucModuleDef used in this Bsw Implementation if it represents a single module several EcucModuleDefs used in this Bsw Implementation if it represents a cluster of modules one or no EcucModuleDefs used in this Bsw Implementation if it represents a library <p>Tags: xml.roleWrapperElement=true</p> |

Table D.10: BswImplementation

| Class | BswInternalBehavior | | | |
|----------------------------|---|-------|------|--|
| Package | M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior | | | |
| Note | Specifies the behavior of a BSW module or a BSW cluster w.r.t. the code entities visible by the BSW Scheduler. It is possible to have several different BswInternalBehaviors referring to the same BswModule Description. | | | |
| Base | ARObject, AtpClassifier, AtpFeature, AtpStructureElement, <i>Identifiable</i> , <i>InternalBehavior</i> , <i>Multilanguage Referrable</i> , <i>Referrable</i> | | | |
| Aggregated by | AtpClassifier.atpFeature, BswModuleDescription.internalBehavior | | | |
| Attribute | Type | Mult. | Kind | Note |
| arTypedPer Instance Memory | VariableDataPrototype | * | agr | <p>Defines an AUTOSAR typed memory-block that needs to be available for each instance of the Basic Software Module. The aggregation of arTypedPerInstanceMemory is subject to variability with the purpose to support variability in the Basic Software Module's implementations. Typically different algorithms in the implementation are requiring different number of memory objects.</p> <p>Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=arTypedPerInstanceMemory.shortName, ar TypedPerInstanceMemory.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p> |





| Class | BswInternalBehavior | | | |
|---------------------------------|---------------------------------|---|------|--|
| bswPerInstanceMemoryPolicy | BswPerInstanceMemoryPolicy | * | aggr | <p>Policy for a arTypedPerInstanceMemory. The policy selects the options of the Schedule Manager API generation.</p> <p>Stereotypes: atpSplitable; atpVariation</p> <p>Tags: atp.Splitkey=bswPerInstanceMemoryPolicy, bswPerInstanceMemoryPolicy.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p> |
| clientPolicy | BswClientPolicy | * | aggr | <p>Policy for a requiredClientServerEntry. The policy selects the options of the Schedule Manager API generation.</p> <p>Stereotypes: atpSplitable; atpVariation</p> <p>Tags: atp.Splitkey=clientPolicy, clientPolicy.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p> |
| distinguishedPartition | BswDistinguishedPartition | * | aggr | <p>Indicates an abstract partition context in which the enclosing BswModuleEntity can be executed.</p> <p>Stereotypes: atpSplitable; atpVariation</p> <p>Tags: atp.Splitkey=distinguishedPartition.shortName, distinguishedPartition.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=60</p> |
| entity | BswModuleEntity | * | aggr | <p>A code entity for which the behavior is described</p> <p>Stereotypes: atpSplitable; atpVariation</p> <p>Tags: atp.Splitkey=entity.shortName, entity.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=5</p> |
| event | BswEvent | * | aggr | <p>An event required by this module behavior.</p> <p>Stereotypes: atpSplitable; atpVariation</p> <p>Tags: atp.Splitkey=event.shortName, event.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=10</p> |
| exclusiveAreaPolicy | BswExclusiveAreaPolicy | * | aggr | <p>Policy for an ExclusiveArea in this BswInternalBehavior. The policy selects the options of the Schedule Manager API generation.</p> <p>Stereotypes: atpSplitable; atpVariation</p> <p>Tags: atp.Splitkey=exclusiveAreaPolicy, exclusiveAreaPolicy.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p> |
| includedDataTypeSet | IncludedDataTypeSet | * | aggr | <p>The includedDataTypeSet is used by a basic software module for its implementation.</p> <p>Stereotypes: atpSplitable</p> <p>Tags: atp.Splitkey=includedDataTypeSet</p> |
| includedModeDeclarationGroupSet | IncludedModeDeclarationGroupSet | * | aggr | <p>This aggregation represents the included Mode DeclarationGroups</p> <p>Stereotypes: atpSplitable</p> <p>Tags: atp.Splitkey=includedModeDeclarationGroupSet</p> |





| Class | BswInternalBehavior | | | |
|---------------------------------|----------------------------------|---|------|---|
| internal TriggeringPoint | BswInternalTriggeringPoint | * | aggr | <p>An internal triggering point.</p> <p>Stereotypes: atpSplitable; atpVariation</p> <p>Tags: <code>atp.Splitkey=internalTriggeringPoint.shortName, internalTriggeringPoint.variationPoint.shortLabel</code> <code>vh.latestBindingTime=preCompileTime</code> <code>xml.sequenceOffset=2</code> </p> |
| internal TriggeringPoint Policy | BswInternalTriggeringPointPolicy | * | aggr | <p>Policy for an internalTriggeringPoint in this BswInternalBehavior.. The policy selects the options of the Schedule Manager API generation.</p> <p>Stereotypes: atpSplitable; atpVariation</p> <p>Tags: <code>atp.Splitkey=internalTriggeringPointPolicy, internalTriggeringPointPolicy.variationPoint.shortLabel</code> <code>vh.latestBindingTime=preCompileTime</code> </p> |
| modeReceiver Policy | BswModeReceiverPolicy | * | aggr | <p>Implementation policy for the reception of mode switches.</p> <p>Stereotypes: atpSplitable; atpVariation</p> <p>Tags: <code>atp.Splitkey=modeReceiverPolicy, modeReceiverPolicy.variationPoint.shortLabel</code> <code>vh.latestBindingTime=preCompileTime</code> <code>xml.sequenceOffset=25</code> </p> |
| modeSender Policy | BswModeSenderPolicy | * | aggr | <p>Implementation policy for providing a mode group.</p> <p>Stereotypes: atpSplitable; atpVariation</p> <p>Tags: <code>atp.Splitkey=modeSenderPolicy, modeSenderPolicy.variationPoint.shortLabel</code> <code>vh.latestBindingTime=preCompileTime</code> <code>xml.sequenceOffset=20</code> </p> |
| parameterPolicy | BswParameterPolicy | * | aggr | <p>Policy for a perInstanceParameter in this BswInternalBehavior. The policy selects the options of the Schedule Manager API generation.</p> <p>Stereotypes: atpSplitable; atpVariation</p> <p>Tags: <code>atp.Splitkey=parameterPolicy, parameterPolicy.variationPoint.shortLabel</code> <code>vh.latestBindingTime=preCompileTime</code> </p> |
| perInstance Parameter | ParameterData Prototype | * | aggr | <p>Describes a read only memory object containing characteristic value(s) needed by this BswInternalBehavior. The role name perInstanceParameter is chosen in analogy to the similar role in the context of SwcInternalBehavior.</p> <p>In contrast to constantMemory, this object is not allocated locally by the module's code, but by the BSW Scheduler and it is accessed from the BSW module via the BSW Scheduler API. The main use case is the support of software emulation of calibration data.</p> <p>The aggregation is subject to variability with the purpose to support implementation variants.</p> <p>Stereotypes: atpSplitable; atpVariation</p> <p>Tags: <code>atp.Splitkey=perInstanceParameter.shortName, perInstanceParameter.variationPoint.shortLabel</code> <code>vh.latestBindingTime=preCompileTime</code> <code>xml.sequenceOffset=45</code> </p> |





| Class | BswInternalBehavior | | | |
|-----------------------------|--------------------------------|---|------|---|
| receptionPolicy | BswDataReceptionPolicy | * | aggr | <p>Data reception policy for inter-partition and/or inter-core communication.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=receptionPolicy, receptionPolicy.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=55</p> |
| releasedTriggerPolicy | BswReleasedTriggerPolicy | * | aggr | <p>Policy for a releasedTrigger. The policy selects the options of the Schedule Manager API generation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=releasedTriggerPolicy, releasedTriggerPolicy.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p> |
| schedulerNamePrefix | BswSchedulerNamePrefix | * | aggr | <p>Optional definition of one or more prefixes to be used for the BswScheduler.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=schedulerNamePrefix.shortName, schedulerNamePrefix.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=50</p> |
| sendPolicy | BswDataSendPolicy | * | aggr | <p>Policy for a providedData. The policy selects the options of the Schedule Manager API generation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=sendPolicy, sendPolicy.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p> |
| serviceDependency | BswServiceDependency | * | aggr | <p>Defines the requirements on AUTOSAR Services for a particular item.</p> <p>The aggregation is subject to variability with the purpose to support the conditional existence of ServiceNeeds.</p> <p>The aggregation is splitable in order to support that ServiceNeeds might be provided in later development steps.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=serviceDependency.ident.shortName, serviceDependency.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=40</p> |
| triggerDirectImplementation | BswTriggerDirectImplementation | * | aggr | <p>Specifies a trigger to be directly implemented via OS calls.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=triggerDirectImplementation, triggerDirectImplementation.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=15</p> |
| variationPointProxy | VariationPointProxy | * | aggr | <p>Proxy of a variation points in the C/C++ implementation.</p> <p>Stereotypes: atpSplitable Tags: atp.Splitkey=variationPointProxy.shortName</p> |

Table D.11: BswInternalBehavior

| | | | | |
|----------------------|---|--------------|-------------|--|
| Class | BswInterruptEntity | | | |
| Package | M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior | | | |
| Note | BSW module entity, which is designed to be triggered by an interrupt. | | | |
| Base | <i>ARObject</i> , <i>BswModuleEntity</i> , <i>ExecutableEntity</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i> | | | |
| Aggregated by | <i>BswInternalBehavior.entity</i> | | | |
| Attribute | Type | Mult. | Kind | Note |
| interruptCategory | BswInterruptCategory | 0..1 | attr | Category of the interrupt |
| interruptSource | String | 0..1 | attr | Allows a textual documentation of the intended interrupt source. |

Table D.12: BswInterruptEntity

| | | | | |
|------------------------|---|--------------|-------------|--|
| Class | BswModuleDescription | | | |
| Package | M2::AUTOSARTemplates::BswModuleTemplate::BswOverview | | | |
| Note | Root element for the description of a single BSW module or BSW cluster. In case it describes a BSW module, the short name of this element equals the name of the BSW module. | | | |
| | Tags: atp.recommendedPackage=BswModuleDescriptions | | | |
| Base | <i>ARElement</i> , <i>ARObject</i> , <i>AtpBlueprint</i> , <i>AtpBlueprintable</i> , <i>AtpClassifier</i> , <i>AtpFeature</i> , <i>AtpStructureElement</i> , <i>CollectableElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>Referrable</i> | | | |
| Aggregated by | <i>ARPackage.element</i> , <i>AtpClassifier.atpFeature</i> | | | |
| Attribute | Type | Mult. | Kind | Note |
| bswModuleDependency | BswModuleDependency | * | aggr | <p>Describes the dependency to another BSW module.</p> <p>Stereotypes: atpSplittable; atpVariation</p> <p>Tags: atp.Splitkey=bswModuleDependency.shortName, bswModuleDependency.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=20 </p> |
| bswModuleDocumentation | SwComponentDocumentation | 0..1 | aggr | <p>This adds a documentation to the BSW module.</p> <p>Stereotypes: atpSplittable; atpVariation</p> <p>Tags: atp.Splitkey=bswModuleDocumentation, bswModuleDocumentation.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=6 </p> |
| expectedEntry | <i>BswModuleEntry</i> | * | ref | <p>Indicates an entry which is required by this module. Replacement of outgoingCallback / requiredEntry.</p> <p>Stereotypes: atpSplittable; atpVariation</p> <p>Tags: atp.Splitkey=expectedEntry.bswModuleEntry, expectedEntry.variationPoint.shortLabel vh.latestBindingTime=preCompileTime </p> |
| implementedEntry | <i>BswModuleEntry</i> | * | ref | <p>Specifies an entry provided by this module which can be called by other modules. This includes "main" functions, interrupt routines, and callbacks. Replacement of providedEntry / expectedCallback.</p> <p>Stereotypes: atpSplittable; atpVariation</p> <p>Tags: atp.Splitkey=implementedEntry.bswModuleEntry, implementedEntry.variationPoint.shortLabel vh.latestBindingTime=preCompileTime </p> |





| Class | BswModuleDescription | | | |
|---------------------------|-------------------------------|------|------|--|
| internalBehavior | BswInternalBehavior | * | aggr | <p>The various BswInternalBehaviors associated with a BswModuleDescription can be distributed over several physical files. Therefore the aggregation is <>atp Splitable>>.</p> <p>Stereotypes: atpSplittable</p> <p>Tags: atp.Splitkey=internalBehavior.shortName xml.sequenceOffset=65</p> |
| moduleId | PositiveInteger | 0..1 | attr | <p>Refers to the BSW Module Identifier defined by the AUTOSAR standard. For non-standardized modules, a proprietary identifier can be optionally chosen.</p> <p>Tags: xml.sequenceOffset=5</p> |
| providedClientServerEntry | BswModuleClientServerEntry | * | aggr | <p>Specifies that this module provides a client server entry which can be called from another partition or core. This entry is declared locally to this context and will be connected to the requiredClientServerEntry of another or the same module via the configuration of the BSW Scheduler.</p> <p>Stereotypes: atpSplittable; atpVariation</p> <p>Tags: atp.Splitkey=providedClientServerEntry.shortName, providedClientServerEntry.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=45</p> |
| providedData | VariableDataPrototype | * | aggr | <p>Specifies a data prototype provided by this module in order to be read from another partition or core. The providedData is declared locally to this context and will be connected to the requiredData of another or the same module via the configuration of the BSW Scheduler.</p> <p>Stereotypes: atpSplittable; atpVariation</p> <p>Tags: atp.Splitkey=providedData.shortName, providedData.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=55</p> |
| providedModeGroup | ModeDeclarationGroupPrototype | * | aggr | <p>A set of modes which is owned and provided by this module or cluster. It can be connected to the required ModeGroups of other modules or clusters via the configuration of the BswScheduler. It can also be synchronized with modes provided via ports by an associated ServiceSwComponentType, EcuAbstractionSwComponentType or ComplexDeviceDriverSwComponentType.</p> <p>Stereotypes: atpSplittable; atpVariation</p> <p>Tags: atp.Splitkey=providedModeGroup.shortName, providedModeGroup.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=25</p> |





| Class | BswModuleDescription | | | |
|---------------------------|-------------------------------|---|------|---|
| releasedTrigger | Trigger | * | aggr | <p>A Trigger released by this module or cluster. It can be connected to the requiredTriggers of other modules or clusters via the configuration of the BswScheduler. It can also be synchronized with Triggers provided via ports by an associated ServiceSwComponentType, Ecu AbstractionSwComponentType or ComplexDeviceDriver SwComponentType.</p> <p>Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=releasedTrigger.shortName, releasedTrigger.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=35</p> |
| requiredClientServerEntry | BswModuleClientServerEntry | * | aggr | <p>Specifies that this module requires a client server entry which can be implemented on another partition or core. This entry is declared locally to this context and will be connected to the providedClientServerEntry of another or the same module via the configuration of the BSW Scheduler.</p> <p>Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=requiredClientServerEntry.shortName, requiredClientServerEntry.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=50</p> |
| requiredData | VariableDataPrototype | * | aggr | <p>Specifies a data prototype required by this module in order to be provided from another partition or core. The required Data is declared locally to this context and will be connected to the providedData of another or the same module via the configuration of the BswScheduler.</p> <p>Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=requiredData.shortName, requiredData.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=60</p> |
| requiredModeGroup | ModeDeclarationGroupPrototype | * | aggr | <p>Specifies that this module or cluster depends on a certain mode group. The requiredModeGroup is local to this context and will be connected to the providedModeGroup of another module or cluster via the configuration of the BswScheduler.</p> <p>Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=requiredModeGroup.shortName, requiredModeGroup.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=30</p> |
| requiredTrigger | Trigger | * | aggr | <p>Specifies that this module or cluster reacts upon an external trigger. This requiredTrigger is declared locally to this context and will be connected to the providedTrigger of another module or cluster via the configuration of the BswScheduler.</p> <p>Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=requiredTrigger.shortName, requiredTrigger.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=40</p> |

Table D.13: BswModuleDescription

| | | | | |
|----------------------|--|--------------|-------------|---|
| Class | BswModuleEntity (abstract) | | | |
| Package | M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior | | | |
| Note | Specifies the smallest code fragment which can be described for a BSW module or cluster within AUTOSAR. | | | |
| Base | <i>ARObject</i> , <i>ExecutableEntity</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i> | | | |
| Subclasses | BswCalledEntity, <i>BswInterruptEntity</i> , BswSchedulableEntity | | | |
| Aggregated by | <i>BswInternalBehavior.entity</i> | | | |
| Attribute | Type | Mult. | Kind | Note |
| accessedModeGroup | ModeDeclarationGroupPrototype | * | ref | <p>A mode group which is accessed via API call by this entity. It shall be a ModeDeclarationGroupPrototype required by this module or cluster.</p> <p>Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=accessedModeGroup.modeDeclarationGroupPrototype, accessedModeGroup.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p> |
| activationPoint | BswInternalTriggeringPoint | * | ref | <p>Activation point used by the module entity to activate one or more internal triggers.</p> <p>Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=activationPoint.bswInternalTriggeringPoint, activationPoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p> |
| callPoint | BswModuleCallPoint | * | aggr | <p>A call point used in the code of this entity.</p> <p>The variability of this association is especially targeted at debug scenarios: It is possible to have one variant calling into the AUTOSAR debug module and another one which doesn't.</p> <p>Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=callPoint.shortName, callPoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p> |
| dataReceivePoint | BswVariableAccess | * | aggr | <p>The data is received via the BSW Scheduler.</p> <p>Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=dataReceivePoint.shortName, dataReceivePoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p> |
| dataSendPoint | BswVariableAccess | * | aggr | <p>The data is sent via the BSW Scheduler.</p> <p>Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=dataSendPoint.shortName, dataSendPoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p> |
| implementedEntry | BswModuleEntry | 0..1 | ref | The entry which is implemented by this module entity. |
| issuedTrigger | Trigger | * | ref | <p>A trigger issued by this entity via BSW Scheduler API call. It shall be a BswTrigger released (i.e. owned) by this module or cluster.</p> <p>Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=issuedTrigger.trigger, issuedTrigger.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p> |





| Class | BswModuleEntity (abstract) | | | |
|---------------------|-----------------------------------|------|-----|--|
| managedModeGroup | ModeDeclarationGroupPrototype | * | ref | <p>A mode group which is managed by this entity. It shall be a ModeDeclarationGroupPrototype provided by this module or cluster.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=managedModeGroup.modeDeclarationGroupPrototype, managedModeGroup.variation Point.shortLabel vh.latestBindingTime=preCompileTime</p> |
| schedulerNamePrefix | BswSchedulerNamePrefix | 0..1 | ref | <p>A prefix to be used in generated names for the BswModuleScheduler in the context of this BswModuleEntity, for example entry point prototypes, macros for dealing with exclusive areas, header file names.</p> <p>Details are defined in the SWS RTE.</p> <p>The prefix supersedes default rules for the prefix of those names.</p> |

Table D.14: BswModuleEntity

| Class | BswModuleEntry | | | |
|--------------------------|---|-------|------|--|
| Package | M2::AUTOSARTemplates::BswModuleTemplate::BswInterfaces | | | |
| Note | <p>This class represents a single API entry (C-function prototype) into the BSW module or cluster.</p> <p>The name of the C-function is equal to the short name of this element with one exception: In case of multiple instances of a module on the same CPU, special rules for "infixes" apply, see description of class BswImplementation.</p> <p>Tags: atp.recommendedPackage=BswModuleEntries</p> | | | |
| Base | ARElement, ARObject, AtpBlueprint, AtpBlueprintable, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable | | | |
| Aggregated by | ARPackage.element | | | |
| Attribute | Type | Mult. | Kind | Note |
| argument (ordered) | SwServiceArg | * | aggr | <p>An argument belonging to this BswModuleEntry.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=argument.shortName, argument.variation Point.shortLabel vh.latestBindingTime=blueprintDerivationTime xml.sequenceOffset=45</p> |
| bswEntryKind | BswEntryKindEnum | 0..1 | attr | <p>This describes whether the entry is concrete or abstract. If the attribute is missing the entry is considered as concrete.</p> <p>Tags: xml.sequenceOffset=40</p> |
| callType | BswCallType | 0..1 | attr | <p>The type of call associated with this service.</p> <p>Tags: xml.sequenceOffset=25</p> |
| executionContext | BswExecutionContext | 0..1 | attr | <p>Specifies the execution context which is required (in case of entries into this module) or guaranteed (in case of entries called from this module) for this service.</p> <p>Tags: xml.sequenceOffset=30</p> |
| functionPrototypeEmitter | NameToken | 0..1 | attr | <p>This attribute is used to control the generation of function prototypes. If set to "RTE", the RTE generates the function prototypes in the Module Interlink Header File.</p> |





| Class | BswModuleEntry | | | |
|---------------------|---------------------|------|------|---|
| isReentrant | Boolean | 0..1 | attr | <p>Reentrancy from the viewpoint of function callers:</p> <ul style="list-style-type: none"> • true: Enables the service to be invoked again, before the service has finished. • false: It is prohibited to invoke the service again before it has finished. <p>Tags: xml.sequenceOffset=15</p> |
| isSynchronous | Boolean | 0..1 | attr | <p>Synchronicity from the viewpoint of function callers:</p> <ul style="list-style-type: none"> • true: This calls a synchronous service, i.e. the service is completed when the call returns. • false: The service (on semantical level) may not be complete when the call returns. <p>Tags: xml.sequenceOffset=20</p> |
| returnType | SwServiceArg | 0..1 | aggr | <p>The return type belonging to this bswModuleEntry.</p> <p>Tags: xml.sequenceOffset=40</p> |
| role | Identifier | 0..1 | attr | <p>Specifies the role of the entry in the given context. It shall be equal to the standardized name of the service call, especially in cases where no ServiceIdentifier is specified, e.g. for callbacks. Note that the ShortName is not always sufficient because it maybe vendor specific (e.g. for callbacks which can have more than one instance).</p> <p>Tags: xml.sequenceOffset=10</p> |
| serviceId | PositiveInteger | 0..1 | attr | <p>Refers to the service identifier of the Standardized Interfaces of AUTOSAR basic software. For non-standardized interfaces, it can optionally be used for proprietary identification.</p> <p>Tags: xml.sequenceOffset=5</p> |
| swServiceImplPolicy | SwServiceImplPolicy | 0..1 | attr | <p>Denotes the implementation policy as a standard function call, inline function or macro. This has to be specified on interface level because it determines the signature of the call.</p> <p>Tags: xml.sequenceOffset=35</p> |

Table D.15: BswModuleEntry

| Class | BswTimingEvent | | | |
|---------------|---|-------|------|--|
| Package | M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior | | | |
| Note | A recurring BswEvent driven by a time period. | | | |
| Base | ARObject, <i>AbstractEvent</i> , <i>BswEvent</i> , <i>BswScheduleEvent</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i> | | | |
| Aggregated by | BswInternalBehavior.event | | | |
| Attribute | Type | Mult. | Kind | Note |
| period | TimeValue | 0..1 | attr | Requirement for the time period (in seconds) by which this event is triggered. |

Table D.16: BswTimingEvent

| Class | ClientServerOperation | | | |
|--------------------|--|-------|------|--|
| Package | M2::AUTOSARTemplates::SWComponentTemplate::PortInterface | | | |
| Note | An operation declared within the scope of a client/server interface. | | | |
| Base | ARObject, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable , MultilanguageReferrable , Referrable | | | |
| Aggregated by | ApplicationInterface.command, AtpClassifier.atpFeature, ClientServerInterface.operation, DiagnosticDataElementInterface.read, DiagnosticDataIdentifierInterface.read, DiagnosticDataIdentifierInterface.write, DiagnosticRoutineInterface.requestResult, DiagnosticRoutineInterface.start, DiagnosticRoutineInterface.stop, PhmRecoveryActionInterface.recovery, ServiceInterface.method | | | |
| Attribute | Type | Mult. | Kind | Note |
| argument (ordered) | ArgumentDataPrototype | * | aggr | <p>An argument of this ClientServerOperation</p> <p>Stereotypes: atpSplittable; atpVariation</p> <p>Tags: atp.Splitkey=argument.shortName, argument.variation Point.shortLabel vh.latestBindingTime=blueprintDerivationTime</p> |
| diagArgIntegrity | Boolean | 0..1 | attr | <p>This attribute shall only be used in the implementation of diagnostic routines to support the case where input and output arguments are allocated in a shared buffer and might unintentionally overwrite input arguments by tentative write operations to output arguments.</p> <p>This situation can happen during sliced execution or while output parameters are arrays (call by reference). The value true means that the ClientServerOperation is aware of the usage of a shared buffer and takes precautions to avoid unintentional overwrite of input arguments.</p> <p>If the attribute does not exist or is set to false the ClientServerOperation does not have to consider the usage of a shared buffer.</p> |
| possibleError | ApplicationError | * | ref | Possible errors that may be raised by the referring operation. |

Table D.17: ClientServerOperation

| Class | ComplexDeviceDriverSwComponentType | | | |
|------------------|--|-------|------|--|
| Package | M2::AUTOSARTemplates::SWComponentTemplate::Components | | | |
| Note | The ComplexDeviceDriverSwComponentType is a special AtomicSwComponentType that has direct access to hardware on an ECU and which is therefore linked to a specific ECU or specific hardware. The ComplexDeviceDriverSwComponentType introduces the possibility to link from the software representation to its hardware description provided by the ECU Resource Template. | | | |
| Tags: | atp.recommendedPackage=SwComponentTypes | | | |
| Base | ARElement, ARObject, AtomicSwComponentType , AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, CollectableElement, Identifiable , MultilanguageReferrable , PackageableElement, Referrable , SwComponentType | | | |
| Aggregated by | ARPackage.element | | | |
| Attribute | Type | Mult. | Kind | Note |
| hardware Element | HwDescriptionEntity | * | ref | Reference from the ComplexDeviceDriverSwComponentType to the description of the used HwElements. |

Table D.18: ComplexDeviceDriverSwComponentType

| | | | | |
|-----------------------------|---|--------------|-------------|---|
| Class | ComponentInCompositionInstanceRef | | | |
| Package | M2::AUTOSARTemplates::SWComponentTemplate::Composition::InstanceRefs | | | |
| Note | The ComponentInCompositionInstanceRef points to a concrete SwComponentPrototype within a CompositionSwComponentType. | | | |
| Base | <i>ARObject</i> , <i>AtpInstanceRef</i> | | | |
| Aggregated by | DiagnosticJ1939SwMapping.swComponentPrototype, <i>EOCEventRef</i> .component, <i>EOCExecutableEntityRef</i> .component, <i>ExecutionTimeConstraint</i> .component, <i>TDEventSwc</i> .component, <i>TDEventVfb</i> .component | | | |
| Attribute | Type | Mult. | Kind | Note |
| base | CompositionSwComponentType | 0..1 | ref | Stereotypes: atpDerived Tags: xml.sequenceOffset=10 |
| context Component (ordered) | SwComponentPrototype | * | ref | The context for the scope of this timing event. Tags: xml.sequenceOffset=20 |
| target Component | SwComponentPrototype | 0..1 | ref | Tags: xml.sequenceOffset=30 |

Table D.19: ComponentInCompositionInstanceRef

| | | | | |
|----------------------|---|--------------|-------------|--|
| Class | CompositionSwComponentType | | | |
| Package | M2::AUTOSARTemplates::SWComponentTemplate::Composition | | | |
| Note | A CompositionSwComponentType aggregates <i>SwComponentPrototypes</i> (that in turn are typed by <i>SwComponentTypes</i>) as well as <i>SwConnectors</i> for primarily connecting <i>SwComponentPrototypes</i> among each others and towards the surface of the CompositionSwComponentType. By this means, a hierarchical structures of software-components can be created. Tags: atp.recommendedPackage=SwComponentTypes | | | |
| Base | <i>ARElement</i> , <i>ARObject</i> , <i>AtpBlueprint</i> , <i>AtpBlueprintable</i> , <i>AtpClassifier</i> , <i>AtpType</i> , <i>CollectableElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>Referrable</i> , <i>SwComponentType</i> | | | |
| Aggregated by | ARPackage.element | | | |
| Attribute | Type | Mult. | Kind | Note |
| component | SwComponentPrototype | * | aggr | <p>The instantiated components that are part of this composition. The aggregation of <i>SwComponentPrototype</i> is subject to variability with the purpose to support the conditional existence of a <i>SwComponentPrototype</i>. Please be aware: if the conditional existence of <i>SwComponentPrototypes</i> is resolved post-build, the deselected <i>SwComponentPrototypes</i> are still contained in the ECUs build but the instances are inactive in that they are not scheduled by the RTE.</p> <p>The aggregation is marked as <i>atpSplittable</i> in order to allow the addition of service components to the ECU extract during the ECU integration.</p> <p>The use case for having 0 components owned by the <i>CompositionSwComponentType</i> could be to deliver an empty <i>CompositionSwComponentType</i> to e.g. a supplier for filling the internal structure.</p> <p>Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=component.shortName, component.variation Point.shortLabel vh.latestBindingTime=postBuild</p> |





| Class | CompositionSwComponentType | | | |
|-----------------------------|----------------------------------|---|------|---|
| connector | SwConnector | * | aggr | <p>SwConnectors have the principal ability to establish a connection among PortPrototypes. They can have many roles in the context of a CompositionSwComponentType. Details are refined by subclasses.</p> <p>The aggregation of SwConnectors is subject to variability with the purpose to support variant data flow.</p> <p>The aggregation is marked as atpSplittable in order to allow the extension of the ECU extract with AssemblySwConnectors between ApplicationSwComponentTypes and ServiceSwComponentTypes during the ECU integration.</p> <p>Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=connector.shortName, connector.variationPoint.shortLabel vh.latestBindingTime=postBuild</p> |
| constantValue Mapping | ConstantSpecification MappingSet | * | ref | <p>Reference to the ConstantSpecificationMapping to be applied for initValues of PPortComSpecs and RPortComSpec.</p> <p>Stereotypes: atpSplittable Tags: atp.Splitkey=constantValueMapping</p> |
| dataType Mapping | DataTypeMappingSet | * | ref | <p>Reference to the DataTypeMappingSet to be applied for the used ApplicationDataTypes in PortInterfaces.</p> <p>Background: when developing subsystems it may happen that ApplicationDataTypes are used on the surface of CompositionSwComponentTypes. In this case it would be reasonable to be able to also provide the intended mapping to the ImplementationDataTypes. However, this mapping shall be informal and not technically binding for the implementors mainly because the RTE generator is not concerned about the CompositionSwComponentTypes.</p> <p>Rationale: if the mapping of ApplicationDataTypes on the delegated and inner PortPrototype matches then the mapping to ImplementationDataTypes is not impacting compatibility.</p> <p>Stereotypes: atpSplittable Tags: atp.Splitkey=dataTypeMapping</p> |
| instantiation RTEEventProps | InstantiationRTEEventProps | * | aggr | <p>This allows to define instantiation specific properties for RTE Events, in particular for instance specific scheduling.</p> <p>Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=instantiationRTEEventProps.shortLabel, instantiationRTEEventProps.variationPoint.shortLabel vh.latestBindingTime=codeGenerationTime</p> |





| Class | CompositionSwComponentType | | | |
|----------------------------|------------------------------|------|-----|--|
| physical Dimension Mapping | PhysicalDimension MappingSet | 0..1 | ref | <p>This reference identifies the <code>PhysicalDimensionMappingSet</code> that is applicable in the context of the enclosing <code>CompositionSwComponentType</code>. The <code>PhysicalDimensionMappings</code> contained in the <code>PhysicalDimensionMappingSet</code> shall be taken into account for the assessment of the compatibility of <code>PhysicalDimensions</code> in the context of creation of a <code>PortInterfaceMapping</code> in the scope of the <code>CompositionSwComponentType</code>.</p> |

Table D.20: CompositionSwComponentType

| Class | CpSoftwareCluster | | | |
|-----------------------|--|-------|------|--|
| Package | M2::AUTOSARTemplates::SystemTemplate::SoftwareCluster | | | |
| Note | <p>This meta class provides the ability to define a CP Software Cluster. Each CP Software Cluster can be integrated and build individually. It defines the sub-set of hierarchical tree(s) of Software Components belonging to this CP Software Cluster. Resources required or provided by this CP Software Cluster are given in the according mappings.</p> <p>Tags: atp.recommendedPackage=CpSoftwareClusters</p> | | | |
| Base | <code>ARElement</code> , <code>ARObject</code> , <code>CollectableElement</code> , <code>Identifiable</code> , <code>MultilanguageReferrable</code> , <code>PackageableElement</code> , <code>Referrable</code> | | | |
| Aggregated by | <code>ARPackage.element</code> | | | |
| Attribute | Type | Mult. | Kind | Note |
| softwareClusterId | PositiveInteger | 0..1 | attr | <p>This attribute represents the value of the id of the corresponding CP software cluster.</p> |
| swComponentAssignment | SwComponent PrototypeAssignment | * | aggr | <p>This is the collection of <code>SwComponentPrototype</code> Assignments</p> <p>Stereotypes: atpSplittable; atpVariation</p> <p>Tags: <code>atp.Splitkey=swComponentAssignment</code>, <code>swComponentAssignment.variationPoint.shortLabel</code> <code>vh.latestBindingTime=postBuild</code> </p> |
| swComposition | CompositionSwComponentType | * | ref | <p>Software Components in the context of a <code>CompositionSwComponentType</code> belonging to this CP Software Cluster. This reference can be used to describe the belonging SWCs when the CP Software Cluster is described out of the context of a System, e.g. reusable CP Software Cluster.</p> <p>Stereotypes: atpSplittable; atpVariation</p> <p>Tags: <code>atp.Splitkey=swComposition.compositionSwComponentType</code>, <code>swComposition.variationPoint.shortLabel</code> <code>vh.latestBindingTime=systemDesignTime</code> </p> |

Table D.21: CpSoftwareCluster

| | | | | |
|----------------------|---|--------------|-------------|---|
| Class | EcuAbstractionSwComponentType | | | |
| Package | M2::AUTOSARTemplates::SWComponentTemplate::Components | | | |
| Note | The ECUAbstraction is a special AtomicSwComponentType that resides between a software-component that wants to access ECU periphery and the Microcontroller Abstraction. The EcuAbstractionSw ComponentType introduces the possibility to link from the software representation to its hardware description provided by the ECU Resource Template. Tags: atp.recommendedPackage=SwComponentTypes | | | |
| Base | <i>ARElement, AROObject, AtomicSwComponentType, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable, SwComponentType</i> | | | |
| Aggregated by | ARPackage.element | | | |
| Attribute | Type | Mult. | Kind | Note |
| hardware Element | HwDescriptionEntity | * | ref | Reference from the EcuAbstractionComponentType to the description of the used HwElements. |

Table D.22: EcuAbstractionSwComponentType

| | | | | |
|----------------------|--|--------------|-------------|--|
| Class | EcucValueCollection | | | |
| Package | M2::AUTOSARTemplates::ECUCDescriptionTemplate | | | |
| Note | This represents the anchor point of the ECU configuration description. Tags: atp.recommendedPackage=EcucValueCollections | | | |
| Base | <i>ARElement, AROObject, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable</i> | | | |
| Aggregated by | ARPackage.element | | | |
| Attribute | Type | Mult. | Kind | Note |
| ecucValue | EcucModule ConfigurationValues | * | ref | References to the configuration of individual software modules that are present on this ECU. atpVariation: [RS_ECUC_00079] Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=ecucValue.ecucModuleConfigurationValues, ecucValue.variationPoint.shortLabel vh.latestBindingTime=preCompileTime |
| ecuExtract | System | 0..1 | ref | Represents the extract of the System Configuration that is relevant for the ECU configured with that ECU Configuration Description. |

Table D.23: EcucValueCollection

| | | | | |
|-------------------|--|--------------|-------------|--|
| Class | ExecutableEntity (abstract) | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::InternalBehavior | | | |
| Note | Abstraction of executable code. | | | |
| Base | <i>ARObject, Identifiable, MultilanguageReferrable, Referrable</i> | | | |
| Subclasses | <i>BswModuleEntity, RunnableEntity</i> | | | |
| Attribute | Type | Mult. | Kind | Note |
| activation Reason | ExecutableEntity ActivationReason | * | aggr | If the ExecutableEntity provides at least one activation Reason element the RTE resp. BSW Scheduler shall provide means to read the activation vector of this executable entity execution. If no activationReason element is provided the feature of being able to determine the activating RTEEvent is disabled for this ExecutableEntity. |





| Class | ExecutableEntity (abstract) | | | |
|---------------------------|------------------------------------|------|------|---|
| canEnter | ExclusiveArea | * | ref | <p>This means that the executable entity can enter/leave the referenced exclusive area through explicit API calls.</p> <p>Stereotypes: atpSplittable; atpVariation</p> <p>Tags: <code>atp.Splitkey=canEnter.exclusiveArea, canEnter.variation</code> <code>Point.shortLabel</code> <code>vh.latestBindingTime=preCompileTime</code> </p> |
| exclusiveAreaNestingOrder | ExclusiveAreaNestingOrder | * | ref | <p>This represents the set of ExclusiveAreaNestingOrders recognized by this ExecutableEntity.</p> |
| minimumStartInterval | TimeValue | 0..1 | attr | <p>Specifies the time in seconds by which two consecutive starts of an ExecutableEntity are guaranteed to be separated.</p> |
| reentrancyLevel | ReentrancyLevelEnum | 0..1 | attr | <p>The reentrancy level of this ExecutableEntity. See the documentation of the enumeration type ReentrancyLevelEnum for details.</p> <p>Please note that nonReentrant interfaces can have also reentrant or multicoreReentrant implementations, and reentrant interfaces can also have multicoreReentrant implementations.</p> |
| runsInside | ExclusiveArea | * | ref | <p>The executable entity runs completely inside the referenced exclusive area.</p> <p>Stereotypes: atpSplittable; atpVariation</p> <p>Tags: <code>atp.Splitkey=runsInside.exclusiveArea, runsInside.variationPoint.shortLabel</code> <code>vh.latestBindingTime=preCompileTime</code> </p> |
| swAddrMethod | SwAddrMethod | 0..1 | ref | <p>Addressing method related to this code entity. Via an association to the same SwAddrMethod, it can be specified that several code entities (even of different modules or components) shall be located in the same memory without already specifying the memory section itself.</p> |

Table D.24: ExecutableEntity

| | |
|------------------|---|
| Primitive | Float |
| Package | M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::PrimitiveTypes |
| Note | <p>An instance of Float is an element from the set of real numbers.</p> <p>Tags: <code>xml.xsd.customType=FLOAT</code> <code>xml.xsd.type=double</code> </p> |

Table D.25: Float

| | |
|-------------------|--|
| Class | «atpMixedString» FormulaExpression (abstract) |
| Package | M2::AUTOSARTemplates::GenericStructure::FormulaLanguage |
| Note | <p>This class represents the syntax of the formula language. The class is modeled as an abstract class in order to be specialized into particular use cases. For each use case the referable objects might be specified in the specialization.</p> |
| Base | <i>ARObject</i> |
| Subclasses | CompuGenericMath, EcucConditionFormula, EcucParameterDerivationFormula, <i>FMFormulaByFeaturesAndAttributes</i> , <i>SwSystemconstDependentFormula</i> , <i>TDEventOccurrenceExpressionFormula</i> , <i>TimingConditionFormula</i> |





| | | | | |
|---------------------|--|--------------|-------------|---|
| Class | «atpMixedString» FormulaExpression (abstract) | | | |
| Attribute | Type | Mult. | Kind | Note |
| atpReference | Referrable | * | ref | <p>The referable object shall yield a numerical / boolean value.</p> <p>Stereotypes: atpAbstract</p> |
| atpString Reference | Referrable | * | ref | <p>The referable object shall yield a string value.</p> <p>Stereotypes: atpAbstract</p> |

 Table D.26: **FormulaExpression**


| | | | | |
|----------------------|---|--------------|-------------|--|
| Class | Frame (abstract) | | | |
| Package | M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication | | | |
| Note | Data frame which is sent over a communication medium. This element describes the pure Layout of a frame sent on a channel. | | | |
| Base | ARObject, CollectableElement, FibexElement, Identifiable , MultilanguageReferrable, Packageable Element, Referrable | | | |
| Subclasses | AbstractEthernetFrame, CanFrame, FlexrayFrame, LinFrame | | | |
| Aggregated by | ARPackage.element | | | |
| Attribute | Type | Mult. | Kind | Note |
| frameLength | Integer | 0..1 | attr | <p>The used length (in bytes) of the referencing frame. Should not be confused with a static byte length reserved for each frame by some platforms (e.g. FlexRay).</p> <p>The frameLength of zero bytes is allowed.</p> <p>Please consider also TPS_SYST_02255.</p> |
| pduToFrame Mapping | PduToFrameMapping | * | aggr | <p>A frames layout as a sequence of Pdus.</p> <p>atpVariation: The content of a frame can be variable.</p> <p>Stereotypes: atpSplittable; atpVariation</p> <p>Tags: atp.Splitkey=pduToFrameMapping.shortName, pduToFrameMapping.variationPoint.shortLabel vh.latestBindingTime=postBuild</p> |

 Table D.27: **Frame**


| | | | | |
|----------------------|---|--------------|-------------|--|
| Class | FrameTriggering (abstract) | | | |
| Package | M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication | | | |
| Note | The FrameTriggering describes the instance of a frame sent on a channel and defines the manner of triggering (timing information) and identification of a frame on the channel, on which it is sent. For the same frame, if FrameTriggerings exist on more than one channel of the same cluster the fan-out/in is handled by the Bus interface. | | | |
| Base | ARObject, Identifiable , MultilanguageReferrable, Referrable | | | |
| Subclasses | CanFrameTriggering, EthernetFrameTriggering, FlexrayFrameTriggering, LinFrameTriggering | | | |
| Aggregated by | PhysicalChannel.frameTriggering | | | |
| Attribute | Type | Mult. | Kind | Note |
| frame | Frame | 0..1 | ref | One frame can be triggered several times, e.g. on different channels. If a frame has no frame triggering, it won't be sent at all. A frame triggering has assigned exactly one frame, which it triggers. |





| Class | <i>FrameTriggering</i> (abstract) | | | |
|---------------|-----------------------------------|---|-----|--|
| framePort | FramePort | * | ref | <p>References to the FramePort on every ECU of the system which sends and/or receives the frame.</p> <p>References for both the sender and the receiver side shall be included when the system is completely defined.</p> |
| pduTriggering | PduTriggering | * | ref | <p>This reference provides the relationship to the Pdu Triggerings that are implemented by the FrameTriggering. The reference is optional since no PduTriggering can be defined for NmPdus and XCP Pdus.</p> <p>Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=pduTriggering.pduTriggering, pduTriggering.variationPoint.shortLabel vh.latestBindingTime=postBuild</p> |

Table D.28: FrameTriggering

| Class | <i>IPdu</i> (abstract) | | | |
|----------------------|--|-------|------|--|
| Package | M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication | | | |
| Note | The IPdu (Interaction Layer Protocol Data Unit) element is used to sum up all Pdus that are routed by the PduR. | | | |
| Base | ARElement, ARObject, CollectableElement, FibexElement, <i>Identifiable</i> , MultilanguageReferrable, PackageableElement, Pdu, <i>Referrable</i> , UploadableDesignElement, UploadablePackageElement | | | |
| Subclasses | ContainerIPdu, DcmIPdu, GeneralPurposeIPdu, ISignalIPdu, J1939DcmIPdu, J1939ProtectedIPdu, MultiplexedIPdu, NPdu, SecuredIPdu, UserDefinedIPdu | | | |
| Aggregated by | ARPackage.element | | | |
| Attribute | Type | Mult. | Kind | Note |
| containedIPduProps | ContainedIPduProps | 0..1 | aggr | Defines whether this IPdu may be collected inside a ContainerIPdu. |

Table D.29: IPdu

| Class | <i>ISignal</i> | | | |
|----------------------|--|-------|------|------|
| Package | M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication | | | |
| Note | <p>Signal of the Interaction Layer. The RTE supports a "signal fan-out" where the same System Signal is sent in different SignalIPdus to multiple receivers.</p> <p>To support the RTE "signal fan-out" each SignalIPdu contains ISignals. If the same System Signal is to be mapped into several SignalIPdus there is one ISignal needed for each ISignalToIPduMapping.</p> <p>ISignals describe the Interface between the Precompile configured RTE and the potentially Postbuild configured Com Stack (see ECUC Parameter Mapping).</p> <p>In case of the SystemSignalGroup an ISignal shall be created for each SystemSignal contained in the SystemSignalGroup.</p> <p>Tags: atp.recommendedPackage=ISignals</p> | | | |
| Base | ARElement, ARObject, CollectableElement, FibexElement, <i>Identifiable</i> , MultilanguageReferrable, PackageableElement, <i>Referrable</i> , UploadableDesignElement, UploadablePackageElement | | | |
| Aggregated by | ARPackage.element | | | |
| Attribute | Type | Mult. | Kind | Note |





| Class | ISignal | | | |
|---------------------|--------------------|------|------|--|
| data Transformation | DataTransformation | 0..1 | ref | <p>Optional reference to a DataTransformation which represents the transformer chain that is used to transform the data that shall be placed inside this ISignal.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=dataTransformation.dataTransformation, dataTransformation.variationPoint.shortLabel vh.latestBindingTime=codeGenerationTime</p> |
| dataTypePolicy | DataTypePolicyEnum | 0..1 | attr | <p>With the aggregation of SwDataDefProps an ISignal specifies how it is represented on the network. This representation follows a particular policy. Note that this causes some redundancy which is intended and can be used to support flexible development methodology as well as subsequent integrity checks.</p> <p>If the policy "networkRepresentationFromComSpec" is chosen the network representation from the ComSpec that is aggregated by the PortPrototype shall be used. If the "override" policy is chosen the requirements specified in the PortInterface and in the ComSpec are not fulfilled by the networkRepresentationProps. In case the System Description doesn't use a complete Software Component Description (VFB View) the "legacy" policy can be chosen.</p> |
| initValue | ValueSpecification | 0..1 | aggr | <p>Optional definition of a ISignal's initialValue in case the System Description doesn't use a complete Software Component Description (VFB View). This supports the inclusion of legacy system signals.</p> <p>This value can be used to configure the Signal's "Init Value".</p> <p>If a full DataMapping exist for the SystemSignal this information may be available from a configured Sender ComSpec and ReceiverComSpec. In this case the initvalues in SenderComSpec and/or ReceiverComSpec override this optional value specification. Further restrictions apply from the RTE specification.</p> |
| iSignalProps | ISignalProps | 0..1 | aggr | <p>Additional optional ISignal properties that may be stored in different files.</p> <p>Stereotypes: atpSplitable Tags: atp.Splitkey=iSignalProps</p> |
| iSignalType | ISignalTypeEnum | 0..1 | attr | This attribute defines whether this ISignal is an array that results in a UINT8_N / UINT8_DYN ComSignalType in the COM configuration or a primitive type. |
| length | UnlimitedInteger | 0..1 | attr | <p>Size of the signal in bits. The size needs to be derived from the mapped VariableDataPrototype according to the mapping of primitive DataTypes to BaseTypes as used in the RTE. Indicates maximum size for dynamic length signals.</p> <p>The ISignal length of zero bits is allowed.</p> |





| Class | ISignal | | | |
|----------------------------------|----------------------------|------|------|--|
| network Representation Props | SwDataDefProps | 0..1 | aggr | <p>Specification of the actual network representation. The usage of SwDataDefProps for this purpose is restricted to the attributes compuMethod and baseType. The optional baseType attributes "memAlignment" and "byteOrder" shall not be used.</p> <p>The attribute "dataTypePolicy" in the SystemTemplate element defines whether this network representation shall be ignored and the information shall be taken over from the network representation of the ComSpec.</p> <p>If "override" is chosen by the system integrator the network representation can violate against the requirements defined in the PortInterface and in the network representation of the ComSpec.</p> <p>In case that the System Description doesn't use a complete Software Component Description (VFB View) this element is used to configure "ComSignalDataInvalid Value" and the Data Semantics.</p> <p>Stereotypes: atpSplittable Tags: atp.Splitkey=networkRepresentationProps</p> |
| reception DefaultValue (ordered) | ValueSpecification | * | aggr | <p>Value used to fill data on the receiver side, if less than expected data is received.</p> <p>The value is expected to cover the entire expected ISignal network payload.</p> |
| systemSignal | SystemSignal | 0..1 | ref | Reference to the System Signal that is supposed to be transmitted in the ISignal. |
| timeout Substitution Value | ValueSpecification | 0..1 | aggr | Defines and enables the ComTimeoutSubstitution for this ISignal. |
| transformation ISignalProps | TransformationISignalProps | * | aggr | <p>A transformer chain consists of an ordered list of transformers. The ISignal specific configuration properties for each transformer are defined in the TransformationISignalProps class. The transformer configuration properties that are common for all ISignals are described in the TransformationTechnology class.</p> <p>Stereotypes: atpSplittable Tags: atp.Splitkey=transformationISignalProps</p> |

Table D.30: ISignal

| Class | ISignalTriggering | | | |
|---------------|--|-------|------|--|
| Package | M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication | | | |
| Note | A ISignalTriggering allows an assignment of ISignals to physical channels. | | | |
| Base | ARObject, <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i> | | | |
| Aggregated by | <i>PhysicalChannel</i> .ISignalTriggering | | | |
| Attribute | Type | Mult. | Kind | Note |
| iSignal | ISignal | 0..1 | ref | This reference shall be used if an ISignal is transported on the PhysicalChannel. This reference forms an XOR relationship with the ISignalTriggering-ISignalGroup reference. |
| iSignalGroup | ISignalGroup | 0..1 | ref | This reference shall be used if an ISignalGroup is transported on the PhysicalChannel. This reference forms an XOR relationship with the ISignal Triggering-ISignal reference. |





| Class | ISignalTriggering | | | |
|-------------|-------------------|---|-----|--|
| iSignalPort | ISignalPort | * | ref | References to the ISignalPort on every ECU of the system which sends and/or receives the ISignal. References for both the sender and the receiver side shall be included when the system is completely defined. |

Table D.31: ISignalTriggering

| | | | | |
|------------|--|--|--|--|
| Class | Identifiable (abstract) | | | |
| Package | M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::Identifiable | | | |
| Note | Instances of this class can be referred to by their identifier (within the namespace borders). In addition to this, Identifiables are objects which contribute significantly to the overall structure of an AUTOSAR description. In particular, Identifiables might contain Identifiables. | | | |
| Base | ARObject, MultilanguageReferrable, Referable | | | |
| Subclasses | ARPackage , AbstractDolpLogicAddressProps , AbstractEvent , AbstractImplementationDataTypeElement , AbstractSecurityEventFilter , AbstractSecurityDsmInstanceFilter , AbstractServiceInstance , AppOsTaskProxyToEcuTaskProxyMapping , ApplicationEndpoint , ApplicationError , ApplicationPartitionToEcuPartitionMapping , AppliedStandard , AsynchronousServerCallResultPoint , AtpBlueprint , AtpBlueprintable , AtpClassifier , AtpFeature , AutosarOperationArgumentInstance , AutosarVariableInstance , BinaryManifestAddressableObject , BinaryManifestItemDefinition , BinaryManifestResource , BinaryManifestResourceDefinition , BlockState , BswInternalTriggeringPoint , BswModuleDependency , BuildActionEntity , BuildActionEnvironment , CanTpAddress , CanTpChannel , CanTpNode , Chapter , ClassContentConditional , ClientIdDefinition , ClientServerOperation , Code , CollectableElement , ComManagementMapping , CommConnectorPort , CommunicationConnector , CommunicationController , Compiler , ConsistencyNeeds , ConsumedEventGroup , CouplingElementAbstractDetails , CouplingPort , CouplingPortAbstractShaper , CouplingPortStructuralElement , CpSoftwareClusterResource , CpSoftwareClusterResourceToApplicationPartitionMapping , CpSoftwareClusterToApplicationPartitionMapping , CpSoftwareClusterToEcuInstanceMapping , CpSoftwareClusterToResourceMapping , CryptoServiceMapping , DataPrototypeGroup , DataPrototypeTransformationPropsId , DataTransformation , DdsCpDomain , DdsCpPartition , DdsCpQosProfile , DdsCpTopic , DependencyOnArtifact , DiagEventDebounceAlgorithm , DiagnosticAuthTransmitCertificateEvaluation , DiagnosticConnectedIndicator , DiagnosticDataElement , DiagnosticDebounceAlgorithmProps , DiagnosticFunctionInhibitSource , DiagnosticParameterElement , DiagnosticRoutineSubfunction , DltApplication , DltArgument , DltLogChannel , DltMessage , DolpInterface , DolpLogicAddress , DolpRoutingActivation , ECUMapping , EOCExecutableEntityRefAbstract , EcuPartition , EcucContainerValue , EcucDefinitionElement , EcucDestinationUriDef , EcucEnumerationLiteralDef , EcucQuery , EcucValidationCondition , EndToEndProtection , EthernetWakeUpSleepOnDatalineConfig , EventHandler , ExclusiveArea , ExecutableEntity , ExecutionTime , FMAtributeDef , FMFeatureMapAssertion , FMFeatureMapCondition , FMFeatureMapElement , FMFeatureRelation , FMFeatureRestriction , FMFeatureSelection , FlatInstanceDescriptor , FlexrayArTpNode , FlexrayTpConnectionControl , FlexrayTpNode , FlexrayTpPduPool , FrameTriggering , GeneralParameter , GlobalTimeGateway , GlobalTimeMaster , GlobalTimeSlave , HeapUsage , HwAttributeDef , HwAttributeLiteralDef , HwPin , HwPinGroup , IEEE1722TpAcfBus , IEEE1722TpAcfBusPart , IPSecRule , IPv6ExtHeaderFilterList , ISignalToIPduMapping , ISignalTriggering , IdentCaption , ImpositionTime , InternalTriggeringPoint , J1939SharedAddressCluster , J1939TpNode , Keyword , LifeCycleState , LinScheduleTable , LinTpNode , Linker , MacAddressVlanMembership , MacMulticastGroup , MacSeckeyParticipant , McDataInstance , MemorySection , ModeDeclaration , ModeDeclarationMapping , ModeSwitchPoint , NetworkEndpoint , NmCluster , NmEcu , NmNode , NvBlockDescriptor , PackageableElement , ParameterAccess , PduActivationRoutingGroup , PduToFrameMapping , PduTriggering , PerInstanceMemory , PhysicalChannel , PortElementToCommunicationResourceMapping , PortGroup , PortInterfaceMapping , ResourceConsumption , RootSwCompositionPrototype , RptComponent , RptContainer , RptExecutableEntity , RptExecutableEntityEvent , RptExecutionContext , RptProfile , RptServicePoint , RteEventInCompositionSeparation , RteEventInCompositionToOsTaskProxyMapping , RunnableEntityGroup , SdgAttribute , SdgClass , SecOcJobRequirement , SecureCommunicationAuthenticationProps , SecureCommunicationFreshnessProps , SecurityEventContextDataElement , SecurityEventContextProps , ServerCallPoint , ServiceNeeds , SignalServiceTranslationElementProps , SignalServiceTranslationEventProps , SignalServiceTranslationProps , SocketAddress , SomeipTpChannel , SpecElementReference , StackUsage , StaticSocketConnection , StructuredReq , SwGenericAxisParamType , SwServiceArg , SwcServiceDependency , SwcToApplicationPartitionMapping , SwcToEcuMapping , SwcToImplMapping , SwitchAsynchronousTrafficShaperGroupEntry , SwitchFlowMeteringEntry , SwitchStreamFilterActionDestPortModification , SwitchStreamFilterEntry , SwitchStreamFilterRule , SwitchStreamGateEntry , Switch | | | |





| Class | <i>Identifiable</i> (abstract) | | | |
|--|--------------------------------|-------|------|--|
| StreamIdentification, SystemMapping, SystemSignalGroupToCommunicationResourceMapping, SystemSignalToCommunicationResourceMapping, TDCpSoftwareClusterMapping , TDCpSoftwareClusterResourceMapping , TcpOptionFilterList, TimingClock , TimingClockSyncAccuracy , TimingCondition , TimingConstraint , TimingDescription , TimingExtensionResource , TimingModelInstance , TlsCryptoCipherSuite, TlsCryptoCipherSuiteProps, Topic1, TpAddress, TraceableTable, TraceableText, TracedFailure , TransformationISignalPropsIdent, TransformationProps , TransformationTechnology, Trigger , VariableAccess , VariationPointProxy, ViewMap, VlanConfig, WaitPoint | | | | |
| Attribute | Type | Mult. | Kind | Note |
| adminData | AdminData | 0..1 | aggr | <p>This represents the administrative data for the identifiable object.</p> <p>Stereotypes: atpSplittable</p> <p>Tags:</p> <p>atp.Splitkey=adminData xml.sequenceOffset=-40</p> |
| annotation | Annotation | * | aggr | <p>Possibility to provide additional notes while defining a model element (e.g. the ECU Configuration Parameter Values). These are not intended as documentation but are mere design notes.</p> <p>Tags: xml.sequenceOffset=-25</p> |
| category | CategoryString | 0..1 | attr | <p>The category is a keyword that specializes the semantics of the Identifiable. It affects the expected existence of attributes and the applicability of constraints.</p> <p>Tags: xml.sequenceOffset=-50</p> |
| desc | MultiLanguageOverviewParagraph | 0..1 | aggr | <p>This represents a general but brief (one paragraph) description what the object in question is about. It is only one paragraph! Desc is intended to be collected into overview tables. This property helps a human reader to identify the object in question.</p> <p>More elaborate documentation, (in particular how the object is built or used) should go to "introduction".</p> <p>Tags: xml.sequenceOffset=-60</p> |
| introduction | DocumentationBlock | 0..1 | aggr | <p>This represents more information about how the object in question is built or is used. Therefore it is a DocumentationBlock.</p> <p>Tags: xml.sequenceOffset=-30</p> |
| uuid | String | 0..1 | attr | <p>The purpose of this attribute is to provide a globally unique identifier for an instance of a meta-class. The values of this attribute should be globally unique strings prefixed by the type of identifier. For example, to include a DCE UUID as defined by The Open Group, the UUID would be preceded by "DCE:". The values of this attribute may be used to support merging of different AUTOSAR models. The form of the UUID (Universally Unique Identifier) is taken from a standard defined by the Open Group (was Open Software Foundation). This standard is widely used, including by Microsoft for COM (GUIDs) and by many companies for DCE, which is based on CORBA. The method for generating these 128-bit IDs is published in the standard and the effectiveness and uniqueness of the IDs is not in practice disputed. If the id namespace is omitted, DCE is assumed. An example is "DCE:2fac1234-31f8-11b4-a222-08002b34c003". The uuid attribute has no semantic meaning for an AUTOSAR model and there is no requirement for AUTOSAR tools to manage the timestamp.</p> <p>Tags: xml.attribute=true</p> |

Table D.32: Identifiable

| | | | | |
|----------------------------|--|--------------|-------------|---|
| Class | <i>InternalBehavior</i> (abstract) | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::InternalBehavior | | | |
| Note | Common base class (abstract) for the internal behavior of both software components and basic software modules/clusters. | | | |
| Base | <i>ARObject</i> , <i>AtpClassifier</i> , <i>AtpFeature</i> , <i>AtpStructureElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i> | | | |
| Subclasses | <i>BswInternalBehavior</i> , <i>SwcInternalBehavior</i> | | | |
| Aggregated by | <i>AtpClassifier</i> . <i>atpFeature</i> | | | |
| Attribute | Type | Mult. | Kind | Note |
| constant Memory | ParameterData Prototype | * | aggr | <p>Describes a read only memory object containing characteristic value(s) implemented by this Internal Behavior.</p> <p>The shortName of ParameterDataPrototype has to be equal to the "C" identifier of the described constant.</p> <p>The characteristic value(s) might be shared between Sw ComponentPrototypes of the same SwComponentType.</p> <p>The aggregation of constantMemory is subject to variability with the purpose to support variability in the software component or module implementations.</p> <p>Typically different algorithms in the implementation are requiring different number of memory objects.</p> <p>Stereotypes: <i>atpSplittable</i>; <i>atpVariation</i></p> <p>Tags: <i>atp.Splitkey=constantMemory.shortName</i>, <i>constantMemory.variationPoint.shortLabel</i> <i>vh.latestBindingTime=preCompileTime</i></p> |
| constantValue Mapping | ConstantSpecification MappingSet | * | ref | <p>Reference to the ConstantSpecificationMapping to be applied for the particular InternalBehavior</p> <p>Stereotypes: <i>atpSplittable</i></p> <p>Tags: <i>atp.Splitkey=constantValueMapping</i></p> |
| dataType Mapping | DataTypeMappingSet | * | ref | <p>Reference to the DataTypeMapping to be applied for the particular InternalBehavior</p> <p>Stereotypes: <i>atpSplittable</i></p> <p>Tags: <i>atp.Splitkey=dataTypeMapping</i></p> |
| exclusiveArea | ExclusiveArea | * | aggr | <p>This specifies an ExclusiveArea for this InternalBehavior. The exclusiveArea is local to the component resp. module. The aggregation of ExclusiveAreas is subject to variability. Note: the number of ExclusiveAreas might vary due to the conditional existence of RunnableEntities or BswModuleEntities.</p> <p>Stereotypes: <i>atpSplittable</i>; <i>atpVariation</i></p> <p>Tags: <i>atp.Splitkey=exclusiveArea.shortName</i>, <i>exclusiveArea.variationPoint.shortLabel</i> <i>vh.latestBindingTime=preCompileTime</i></p> |
| exclusiveArea NestingOrder | ExclusiveAreaNesting Order | * | aggr | <p>This represents the set of ExclusiveAreaNestingOrder owned by the InternalBehavior.</p> <p>Stereotypes: <i>atpSplittable</i>; <i>atpVariation</i></p> <p>Tags: <i>atp.Splitkey=exclusiveAreaNestingOrder.shortName</i>, <i>exclusiveAreaNestingOrder.variationPoint.shortLabel</i> <i>vh.latestBindingTime=preCompileTime</i></p> |





| Class | <i>InternalBehavior</i> (abstract) | | | |
|--------------|------------------------------------|---|------|---|
| staticMemory | VariableDataPrototype | * | aggr | <p>Describes a read and writeable static memory object representing measurement variables implemented by this software component. The term "static" is used in the meaning of "non-temporary" and does not necessarily specify a linker encapsulation. This kind of memory is only supported if supportsMultipleInstantiation is FALSE.</p> <p>The shortName of the VariableDataPrototype has to be equal with the "C" identifier of the described variable.</p> <p>The aggregation of staticMemory is subject to variability with the purpose to support variability in the software component's implementations.</p> <p>Typically different algorithms in the implementation are requiring different number of memory objects.</p> <p>Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=staticMemory.shortName, static Memory.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p> |

Table D.33: InternalBehavior

| Class | <i>ModeDeclaration</i> | | | |
|---------------|--|-------|------|--|
| Package | M2::AUTOSARTemplates::CommonStructure::ModeDeclaration | | | |
| Note | Declaration of one Mode. The name and semantics of a specific mode is not defined in the meta-model. | | | |
| Base | ARObject, AtpClassifier, AtpFeature, AtpStructureElement, <i>Identifiable</i> , MultilanguageReferrable, <i>Referrable</i> | | | |
| Aggregated by | AtpClassifier.atpFeature, ModeDeclarationGroup.modeDeclaration | | | |
| Attribute | Type | Mult. | Kind | Note |
| value | PositiveInteger | 0..1 | attr | The RTE shall take the value of this attribute for generating the source code representation of this Mode Declaration. |

Table D.34: ModeDeclaration

| Class | <i>ModeDeclarationGroupPrototype</i> | | | |
|---------------------|--|-------|------|---|
| Package | M2::AUTOSARTemplates::CommonStructure::ModeDeclaration | | | |
| Note | The ModeDeclarationGroupPrototype specifies a set of Modes (ModeDeclarationGroup) which is provided or required in the given context. | | | |
| Base | ARObject, AtpFeature, AtpPrototype, <i>Identifiable</i> , MultilanguageReferrable, <i>Referrable</i> | | | |
| Aggregated by | AtpClassifier.atpFeature, BswModuleDescription.providedModeGroup, BswModuleDescription.requiredModeGroup, FirewallStateSwitchInterface.firewallStateMachine, FunctionGroupSet.functionGroup, ModeSwitchInterface.modeGroup, Process.processStateMachine, StateManagementStateNotification.stateMachine | | | |
| Attribute | Type | Mult. | Kind | Note |
| swCalibrationAccess | SwCalibrationAccess Enum | 0..1 | attr | This allows for specifying whether or not the enclosing ModeDeclarationGroupPrototype can be measured at run-time. |
| type | ModeDeclarationGroup | 0..1 | tref | The "collection of ModeDeclarations" (= ModeDeclaration Group) supported by a component Stereotypes: isOfType |

Table D.35: ModeDeclarationGroupPrototype

| | |
|------------------|--|
| Primitive | Numerical |
| Package | M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::PrimitiveTypes |
| Note | <p>This primitive specifies a numerical value. It can be denoted in different formats such as Decimal, Octal, Hexadecimal, Float. See the xsd pattern for details.</p> <p>The value can be expressed in octal, hexadecimal, binary representation. Negative numbers can only be expressed in decimal or float notation.</p> <p>Tags:</p> <pre>xml.xsd.customType=NUMERICAL-VALUE xml.xsd.pattern=(0[xX][0-9a-fA-F]+) (0[0-7]+) (0[bB][0-1]+) (([+\-]?[1-9][0-9]+ ([0-9]+ ([0-9]+)? [+\-]?[0-9]([0-9]+)?)([eE]([+\-]?[0-9]+)? \.[0-9]+ INF -INF NaN) xml.xsd.type=string</pre> |

Table D.36: Numerical

| | | | | |
|----------------------|--|--------------|-------------|---|
| Class | PPortPrototype | | | |
| Package | M2::AUTOSARTemplates::SWComponentTemplate::Components | | | |
| Note | Component port providing a certain port interface. | | | |
| Base | <i>ARObject, AbstractProvidedPortPrototype, AtpBlueprintable, AtpFeature, AtpPrototype, Identifiable, MultilanguageReferrable, PortPrototype, Referrable</i> | | | |
| Aggregated by | <i>AtpClassifier.atpFeature, SwComponentType.port</i> | | | |
| Attribute | Type | Mult. | Kind | Note |
| provided Interface | PortInterface | 0..1 | ref | <p>The interface that this port provides.</p> <p>Stereotypes: isOfType</p> |

Table D.37: PPortPrototype

| | | | | |
|----------------------|---|--------------|-------------|--|
| Class | PduTriggering | | | |
| Package | M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication | | | |
| Note | <p>The PduTriggering describes on which channel the IPdu is transmitted. The Pdu routing by the PduR is only allowed for subclasses of IPdu.</p> <p>Depending on its relation to entities such channels and clusters it can be unambiguously deduced whether a fan-out is handled by the Pdu router or the Bus Interface.</p> <p>If the fan-out is specified between different clusters it shall be handled by the Pdu Router. If the fan-out is specified between different channels of the same cluster it shall be handled by the Bus Interface.</p> | | | |
| Base | <i>ARObject, Identifiable, MultilanguageReferrable, Referrable</i> | | | |
| Aggregated by | <i>PhysicalChannel.pduTriggering</i> | | | |
| Attribute | Type | Mult. | Kind | Note |
| iPdu | Pdu | 0..1 | ref | <p>Reference to the Pdu for which the PduTriggering is defined. One I-Pdu can be triggered on different channels (PduR fan-out). The Pdu routing by the PduR is only allowed for subclasses of IPdu.</p> <p>Nevertheless is the reference to the Pdu element necessary since the PduTriggering element is also used to specify the sending and receiving connections to Ecu Ports.</p> |
| iPduPort | IPduPort | * | ref | <p>References to the IPduPort on every ECU of the system which sends and/or receives the I-PDU.</p> <p>References for both the sender and the receiver side shall be included when the system is completely defined.</p> |





| Class | PduTriggering | | | |
|---------------------------|----------------------------|------|------|--|
| iSignal Triggering | ISignalTriggering | * | ref | <p>This reference provides the relationship to the ISignal Triggerings that are implemented by the PduTriggering. The reference is optional since no ISignalTriggering can be defined for DCM and Multiplexed Pdus.</p> <p>Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=iSignalTriggering.iSignalTriggering, iSignal Triggering.variationPoint.shortLabel vh.latestBindingTime=postBuild</p> |
| secOcCrypto Mapping | SecOcCryptoService Mapping | 0..1 | ref | <p>This reference identifies the crypto profile applicable to the usage (send, receive) of the also referenced Secured IPdu.</p> <p>Obviously, this reference is only applicable if the Pdutriggering also references a SecuredIPdu in the role i Pdu.</p> |
| triggerIPduSend Condition | TriggerIPduSend Condition | * | aggr | <p>Defines the trigger for the Com_TriggerIPDUSend API call. Only if all defined TriggerIPduSendConditions evaluate to true (AND associated) the Com_Trigger IPDUSend API shall be called.</p> |

Table D.38: PduTriggering

| Class | PhysicalChannel (abstract) | | | |
|----------------|---|-------|------|--|
| Package | M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology | | | |
| Note | <p>A physical channel is the transmission medium that is used to send and receive information between communicating ECUs. Each CommunicationCluster has at least one physical channel. Bus systems like CAN and LIN only have exactly one PhysicalChannel. A FlexRay cluster may have more than one PhysicalChannels that may be used in parallel for redundant communication.</p> <p>An ECU is part of a cluster if it contains at least one controller that is connected to at least one channel of the cluster.#</p> | | | |
| Base | ARObject, Identifiable , MultilanguageReferrable , Referrable | | | |
| Subclasses | AbstractCanPhysicalChannel, EthernetPhysicalChannel, FlexrayPhysicalChannel, LinPhysicalChannel, UserDefinedPhysicalChannel | | | |
| Aggregated by | CommunicationCluster.physicalChannel | | | |
| Attribute | Type | Mult. | Kind | Note |
| comm Connector | Communication Connector | * | ref | <p>Reference to the ECUInstance via a Communication Connector to which the channel is connected.</p> <p>atpVariation: Variable assignment of Physical Channels to different CommunicationConnectors is expressed with this variation.</p> <p>Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=commConnector.communicationConnector, commConnector.variationPoint.shortLabel vh.latestBindingTime=postBuild</p> |





| Class | <i>PhysicalChannel</i> (abstract) | | | |
|--------------------------|-----------------------------------|---|------|--|
| frameTriggering | FrameTriggering | * | aggr | <p>One frame triggering is defined for exactly one channel. Channels may have assigned an arbitrary number of frame triggerings.</p> <p>atpVariation: If signals/PDUs/frames are variable, the corresponding triggerings shall be variable, too.</p> <p>Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=frameTriggering.shortName, frameTriggering.variationPoint.shortLabel vh.latestBindingTime=postBuild</p> |
| iSignal Triggering | ISignalTriggering | * | aggr | <p>One ISignalTriggering is defined for exactly one channel. Channels may have assigned an arbitrary number of ISignaltriggerings.</p> <p>atpVariation: If signals/PDUs/frames are variable, the corresponding triggerings shall be variable, too.</p> <p>Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=iSignalTriggering.shortName, iSignalTriggering.variationPoint.shortLabel vh.latestBindingTime=postBuild</p> |
| managed Physical Channel | PhysicalChannel | * | ref | Reference between a channel with role managing channel and a channel with role managed channel. |
| pduTriggering | PduTriggering | * | aggr | <p>One PduTriggering is defined for exactly one channel. Channels may have assigned an arbitrary number of I-Pdu triggerings.</p> <p>atpVariation: If signals/PDUs/frames are variable, the corresponding triggerings shall be variable, too.</p> <p>Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=pduTriggering.shortName, pduTriggering.variationPoint.shortLabel vh.latestBindingTime=postBuild</p> |

Table D.39: PhysicalChannel

| Class | <i>PortInterface</i> (abstract) | | | |
|----------------------|--|--------------|-------------|-------------|
| Package | M2::AUTOSARTemplates::SWComponentTemplate::PortInterface | | | |
| Note | Abstract base class for an interface that is either provided or required by a port of a software component. | | | |
| Base | <i>ARElement, AROObject, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable</i> | | | |
| Subclasses | ClientServerInterface, DataInterface, ModeSwitchInterface, TriggerInterface | | | |
| Aggregated by | ARPackage.element | | | |
| Attribute | Type | Mult. | Kind | Note |





| Class | PortInterface (abstract) | | | |
|-------------|---------------------------------|------|------|---|
| isService | Boolean | 0..1 | attr | <p>This flag is set if the PortInterface is to be used for communication between an</p> <ul style="list-style-type: none"> • ApplicationSwComponentType or • ServiceProxySwComponentType or • SensorActuatorSwComponentType or • ComplexDeviceDriverSwComponentType • ServiceSwComponentType • EcuAbstractionSwComponentType <p>and a ServiceSwComponentType (namely an AUTOSAR Service) located on the same ECU. Otherwise the flag is not set.</p> <p>Stereotypes: atpVariation Tags: vh.latestBindingTime=blueprintDerivationTime</p> |
| serviceKind | ServiceProviderEnum | 0..1 | attr | This attribute provides further details about the nature of the applied service. |

Table D.40: PortInterface

| Class | PortPrototype (abstract) | | | |
|---------------------------------|--|-------|------|---|
| Package | M2::AUTOSARTemplates::SWComponentTemplate::Components | | | |
| Note | <p>Base class for the ports of an AUTOSAR software component.</p> <p>The aggregation of PortPrototypes is subject to variability with the purpose to support the conditional existence of ports.</p> | | | |
| Base | ARObject, AtpBlueprintable, AtpFeature, AtpPrototype, Identifiable , MultilanguageReferrable, Referrable | | | |
| Subclasses | AbstractProvidedPortPrototype , AbstractRequiredPortPrototype | | | |
| Aggregated by | AtpClassifier.atpFeature , SwComponentType.port | | | |
| Attribute | Type | Mult. | Kind | Note |
| clientServerAnnotation | ClientServerAnnotation | * | aggr | Annotation of this PortPrototype with respect to client/server communication. |
| delegatedPortAnnotation | DelegatedPortAnnotation | 0..1 | aggr | Annotations on this delegated port. |
| ioHwAbstractionServerAnnotation | IoHwAbstractionServerAnnotation | * | aggr | Annotations on this IO Hardware Abstraction port. |
| modePortAnnotation | ModePortAnnotation | * | aggr | Annotations on this mode port. |
| nvDataPortAnnotation | NvDataPortAnnotation | * | aggr | Annotations on this non volatile data port. |
| parameterPortAnnotation | ParameterPortAnnotation | * | aggr | Annotations on this parameter port. |
| senderReceiverAnnotation | SenderReceiverAnnotation | * | aggr | Collection of annotations of this ports sender/receiver communication. |
| triggerPortAnnotation | TriggerPortAnnotation | * | aggr | Annotations on this trigger port. |

Table D.41: PortPrototype

| | | | | |
|----------------------|--|--------------|-------------|--|
| Class | PortPrototypeBlueprint | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::StandardizationTemplate::BlueprintDedicated::Port PrototypeBlueprint | | | |
| Note | This meta-class represents the ability to express a blueprint of a PortPrototype by referring to a particular PortInterface. This blueprint can then be used as a guidance to create particular PortPrototypes which are defined according to this blueprint. By this it is possible to standardize application interfaces without the need to also standardize software-components with PortPrototypes typed by the standardized Port Interfaces. | | | |
| | Tags: atp.recommendedPackage=PortPrototypeBlueprints | | | |
| Base | ARElement, ARObject, AtpBlueprint, AtpClassifier, AtpFeature, AtpStructureElement, Collectable Element, <i>Identifiable</i> , MultilanguageReferrable, PackageableElement, <i>Referrable</i> | | | |
| Aggregated by | ARPackage.element, AtpClassifier.atpFeature | | | |
| Attribute | Type | Mult. | Kind | Note |
| initValue | PortPrototypeBlueprint InitValue | * | aggr | This specifies the init values for the dataElements in the particular PortPrototypeBlueprint. |
| interface | PortInterface | 1 | ref | This is the interface for which the blueprint is defined. It may be a blueprint itself or a standardized PortInterface |
| providedCom Spec | PPortComSpec | * | aggr | Provided communication attributes per interface element (data element or operation). |
| requiredCom Spec | RPortComSpec | * | aggr | Required communication attributes, one for each interface element. |

Table D.42: PortPrototypeBlueprint

| | | | | |
|----------------------|--|--------------|-------------|--|
| Class | RPortPrototype | | | |
| Package | M2::AUTOSARTemplates::SWComponentTemplate::Components | | | |
| Note | Component port requiring a certain port interface. | | | |
| Base | ARObject, <i>AbstractRequiredPortPrototype</i> , AtpBlueprintable, AtpFeature, AtpPrototype, <i>Identifiable</i> , MultilanguageReferrable, <i>PortPrototype</i> , <i>Referrable</i> | | | |
| Aggregated by | AtpClassifier.atpFeature, <i>SwComponentType</i> .port | | | |
| Attribute | Type | Mult. | Kind | Note |
| mayBe Unconnected | Boolean | 0..1 | attr | If set to true, this attribute indicates that the enclosing RPortPrototype may be left unconnected and that this aspect has explicitly been considered in the software-component's design. |
| required Interface | PortInterface | 0..1 | tref | The interface that this port requires. Stereotypes: isOfType |

Table D.43: RPortPrototype

| | | | | |
|----------------------|--|--------------|-------------|-------------|
| Class | RTEEvent (abstract) | | | |
| Package | M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents | | | |
| Note | Abstract base class for all RTE-related events | | | |
| Base | ARObject, <i>AbstractEvent</i> , AtpClassifier, AtpFeature, AtpStructureElement, <i>Identifiable</i> , Multilanguage Referrable, <i>Referrable</i> | | | |
| Subclasses | AsynchronousServerCallReturnsEvent, BackgroundEvent, DataReceiveErrorEvent, DataReceivedEvent, DataSendCompletedEvent, DataWriteCompletedEvent, ExternalTriggerOccurredEvent, InitEvent, InternalTriggerOccurredEvent, ModeSwitchedAckEvent, OperationInvokedEvent, OsTaskExecutionEvent, SwcModeManagerErrorEvent, SwcModeSwitchEvent, <i>TimingEvent</i> , TransformerHardErrorEvent | | | |
| Aggregated by | AtpClassifier.atpFeature, <i>SwcInternalBehavior</i> .event | | | |
| Attribute | Type | Mult. | Kind | Note |





| Class | <i>RTEEvent</i> (abstract) | | | |
|--------------|----------------------------|------|------|--|
| disabledMode | ModeDeclaration | * | iref | <p>Reference to the Modes that disable the Event.</p> <p>Stereotypes: atpSplittable</p> <p>Tags: atp.Splitkey=disabledMode.contextPort, disabledMode.contextModeDeclarationGroupPrototype, disabledMode.targetModeDeclaration</p> <p>InstanceRef implemented by: RModelInAtomicSwcInstanceRef</p> |
| startOnEvent | RunnableEntity | 0..1 | ref | The referenced RunnableEntity starts when the corresponding RTEEvent is raised. |

Table D.44: RTEEvent

| Class | <i>Referrable</i> (abstract) | | | |
|-------------------|--|-------|------|--|
| Package | M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::Identifiable | | | |
| Note | Instances of this class can be referred to by their identifier (while adhering to namespace borders). | | | |
| Base | ARObject | | | |
| Subclasses | <i>AtpDefinition</i> , <i>BswDistinguishedPartition</i> , <i>BswModuleCallPoint</i> , <i>BswModuleClientServerEntry</i> , <i>BswVariableAccess</i> , <i>CouplingPortTrafficClassAssignment</i> , <i>DiagnosticEnvModeElement</i> , <i>EthernetPriorityRegeneration</i> , <i>ExclusiveAreaNestingOrder</i> , <i>HwDescriptionEntity</i> , <i>ImplementationProps</i> , <i>LinSlaveConfigIdent</i> , <i>ModeTransition</i> , <i>MultilanguageReferrable</i> , <i>PncMappingIdent</i> , <i>SingleLanguageReferrable</i> , <i>SoConlPdulIdentifier</i> , <i>SocketConnectionBundle</i> , <i>TimeSyncServerConfiguration</i> , <i>TpConnectionIdent</i> | | | |
| Attribute | Type | Mult. | Kind | Note |
| shortName | Identifier | 1 | attr | <p>This specifies an identifying shortName for the object. It needs to be unique within its context and is intended for humans but even more for technical reference.</p> <p>Stereotypes: atpIdentityContributor</p> <p>Tags:</p> <ul style="list-style-type: none"> xml.enforceMinMultiplicity=true xml.sequenceOffset=-100 |
| shortNameFragment | ShortNameFragment | * | aggr | <p>This specifies how the Referrable.shortName is composed of several shortNameFragments.</p> <p>Tags: xml.sequenceOffset=-90</p> |

Table D.45: Referrable

| Class | <i>ResourceConsumption</i> | | | |
|----------------------|--|-------|------|---|
| Package | M2::AUTOSARTemplates::CommonStructure::ResourceConsumption | | | |
| Note | Description of consumed resources by one implementation of a software. | | | |
| Base | ARObject, <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i> | | | |
| Aggregated by | <i>EcuResourceEstimation.bswResourceEstimation</i> , <i>EcuResourceEstimation.rteResourceEstimation</i> , <i>Implementation.resourceConsumption</i> , <i>StateDependentStartupConfig.resourceConsumption</i> | | | |
| Attribute | Type | Mult. | Kind | Note |
| accessCountSet | AccessCountSet | * | aggr | <p>Set of access count values</p> <p>Stereotypes: atpSplittable; atpVariation</p> <p>Tags:</p> <ul style="list-style-type: none"> atp.Splitkey=accessCountSet, accessCountSet.variation Point.shortLabel vh.latestBindingTime=preCompileTime |





| Class | ResourceConsumption | | | |
|-------------------|---------------------|---|------|---|
| executionTime | ExecutionTime | * | aggr | <p>Collection of the execution time descriptions for this implementation. The aggregation of executionTime is subject to variability with the purpose to support the conditional existence of runnable entities.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: <code>atp.Splitkey=executionTime.shortName, executionTime.variationPoint.shortLabel</code> <code>vh.latestBindingTime=preCompileTime</code></p> |
| heapUsage | HeapUsage | * | aggr | <p>Collection of the heap memory allocated by this implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: <code>atp.Splitkey=heapUsage.shortName, heapUsage.variationPoint.shortLabel</code> <code>vh.latestBindingTime=preCompileTime</code></p> |
| memorySection | MemorySection | * | aggr | <p>An abstract memory section required by this implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: <code>atp.Splitkey=memorySection.shortName, memorySection.variationPoint.shortLabel</code> <code>vh.latestBindingTime=preCompileTime</code></p> |
| sectionNamePrefix | SectionNamePrefix | * | aggr | <p>A prefix to be used for the memory section symbol in the code.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: <code>atp.Splitkey=sectionNamePrefix.shortName, sectionNamePrefix.variationPoint.shortLabel</code> <code>vh.latestBindingTime=preCompileTime</code></p> |
| stackUsage | StackUsage | * | aggr | <p>Collection of the stack memory usage for each runnable entity of this implementation. The aggregation of Stack Usage is subject to variability with the purpose to support the conditional existence of runnable entities.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: <code>atp.Splitkey=stackUsage.shortName, stackUsage.variationPoint.shortLabel</code> <code>vh.latestBindingTime=preCompileTime</code></p> |

Table D.46: ResourceConsumption

| | | | | |
|--------------------------------|--|--------------|-------------|---|
| Class | RootSwCompositionPrototype | | | |
| Package | M2::AUTOSARTemplates::SystemTemplate | | | |
| Note | <p>The RootSwCompositionPrototype represents the top-level-composition of software components within a given System.</p> <p>According to the use case of the System, this may for example be a more or less complete VFB description, the software of a System Extract or the software of a flat ECU Extract with only atomic SWCs.</p> <p>Therefore the RootSwComposition will only occasionally contain all atomic software components that are used in a complete VFB System. The OEM is primarily interested in the required functionality and the interfaces defining the integration of the Software Component into the System. The internal structure of such a component contains often substantial intellectual property of a supplier. Therefore a top-level software composition will often contain empty compositions which represent subsystems.</p> <p>The contained SwComponentPrototypes are fully specified by their SwComponentTypes (including Port Prototypes, PortInterfaces, VariableDataPrototypes, SwInternalBehavior etc.), and their ports are interconnected using SwConnectorPrototypes.</p> | | | |
| Base | ARObject, AtpFeature, AtpPrototype, Identifiable , MultilanguageReferrable , Referrable | | | |
| Aggregated by | AtpClassifier.atpFeature, System.rootSoftwareComposition | | | |
| Attribute | Type | Mult. | Kind | Note |
| calibration ParameterValue Set | CalibrationParameter ValueSet | * | ref | <p>Used CalibrationParameterValueSet for instance specific initialization of calibration parameters.</p> <p>Stereotypes: atpSplittable</p> <p>Tags: atp.Splitkey=calibrationParameterValueSet</p> |
| flatMap | FlatMap | 0..1 | ref | <p>The FlatMap used in the scope of this RootSw CompositionPrototype.</p> <p>Stereotypes: atpSplittable</p> <p>Tags: atp.Splitkey=flatMap</p> |
| software Composition | CompositionSw ComponentType | 0..1 | tref | <p>We assume that there is exactly one top-level composition that includes all Component instances of the system.</p> <p>Stereotypes: isOfType</p> |

Table D.47: RootSwCompositionPrototype

| | | | | |
|----------------------|---|--------------|-------------|---|
| Class | RunnableEntity | | | |
| Package | M2::AUTOSARTemplates::SWComponentTemplate::SwInternalBehavior | | | |
| Note | <p>A RunnableEntity represents the smallest code-fragment that is provided by an AtomicSwComponent Type and are executed under control of the RTE. RunnableEntities are for instance set up to respond to data reception or operation invocation on a server.</p> | | | |
| Base | ARObject, AtpClassifier, AtpFeature, AtpStructureElement, ExecutableEntity , Identifiable , Multilanguage Referrable , Referrable | | | |
| Aggregated by | AtpClassifier.atpFeature, SwInternalBehavior.runnable | | | |
| Attribute | Type | Mult. | Kind | Note |
| argument (ordered) | RunnableEntity Argument | * | aggr | This represents the formal definition of a an argument to a RunnableEntity. |





| Class | RunnableEntity | | | |
|-------------------------------------|-----------------------------------|------|------|---|
| asynchronous ServerCall ResultPoint | AsynchronousServerCallResultPoint | * | aggr | <p>The server call result point admits a runnable to fetch the result of an asynchronous server call.</p> <p>The aggregation of AsynchronousServerCallResultPoint is subject to variability with the purpose to support the conditional existence of client server PortPrototypes and the variant existence of server call result points in the implementation.</p> <p>Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=asynchronousServerCallResultPoint.shortName, asynchronousServerCallResultPoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p> |
| canBeInvoked Concurrently | Boolean | 0..1 | attr | <p>If the value of this attribute is set to "true" the enclosing RunnableEntity can be invoked concurrently (even for one instance of the corresponding AtomicSwComponent Type). This implies that it is the responsibility of the implementation of the RunnableEntity to take care of this form of concurrency.</p> |
| dataRead Access | VariableAccess | * | aggr | <p>RunnableEntity has implicit read access to dataElement of a sender-receiver PortPrototype or nv data of a nv data PortPrototype.</p> <p>The aggregation of dataReadAccess is subject to variability with the purpose to support the conditional existence of sender receiver ports or the variant existence of dataReadAccess in the implementation.</p> <p>Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=dataReadAccess.shortName, dataReadAccess.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p> |
| dataReceive PointBy Argument | VariableAccess | * | aggr | <p>RunnableEntity has explicit read access to dataElement of a sender-receiver PortPrototype or nv data of a nv data PortPrototype. The result is passed back to the application by means of an argument in the function signature.</p> <p>The aggregation of dataReceivePointByArgument is subject to variability with the purpose to support the conditional existence of sender receiver PortPrototype or the variant existence of data receive points in the implementation.</p> <p>Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=dataReceivePointByArgument.shortName, dataReceivePointByArgument.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p> |





| Class | RunnableEntity | | | |
|-------------------------|-------------------------|---|------|---|
| dataReceivePointByValue | VariableAccess | * | aggr | <p>RunnableEntity has explicit read access to dataElement of a sender-receiver PortPrototype or nv data of a nv data PortPrototype.</p> <p>The result is passed back to the application by means of the return value. The aggregation of dataReceivePointByValue is subject to variability with the purpose to support the conditional existence of sender receiver ports or the variant existence of data receive points in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: <code>atp.Splitkey=dataReceivePointByValue.shortName, dataReceivePointByValue.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</code></p> |
| dataSendPoint | VariableAccess | * | aggr | <p>RunnableEntity has explicit write access to dataElement of a sender-receiver PortPrototype or nv data of a nv data PortPrototype.</p> <p>The aggregation of dataSendPoint is subject to variability with the purpose to support the conditional existence of sender receiver PortPrototype or the variant existence of data send points in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: <code>atp.Splitkey=dataSendPoint.shortName, dataSendPoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</code></p> |
| dataWriteAccess | VariableAccess | * | aggr | <p>RunnableEntity has implicit write access to dataElement of a sender-receiver PortPrototype or nv data of a nv data PortPrototype.</p> <p>The aggregation of dataWriteAccess is subject to variability with the purpose to support the conditional existence of sender receiver ports or the variant existence of dataWriteAccess in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: <code>atp.Splitkey=dataWriteAccess.shortName, dataWriteAccess.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</code></p> |
| externalTriggeringPoint | ExternalTriggeringPoint | * | aggr | <p>The aggregation of ExternalTriggeringPoint is subject to variability with the purpose to support the conditional existence of trigger ports or the variant existence of external triggering points in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: <code>atp.Splitkey=externalTriggeringPoint.ident.shortName, externalTriggeringPoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</code></p> |
| internalTriggeringPoint | InternalTriggeringPoint | * | aggr | <p>The aggregation of InternalTriggeringPoint is subject to variability with the purpose to support the variant existence of internal triggering points in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: <code>atp.Splitkey=internalTriggeringPoint.shortName, internalTriggeringPoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</code></p> |





| Class | RunnableEntity | | | |
|--------------------|-----------------|---|------|---|
| modeAccess Point | ModeAccessPoint | * | aggr | <p>The runnable has a mode access point. The aggregation of ModeAccessPoint is subject to variability with the purpose to support the conditional existence of mode ports or the variant existence of mode access points in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: <code>atp.Splitkey=modeAccessPoint.ident.shortName, modeAccessPoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</code></p> |
| modeSwitch Point | ModeSwitchPoint | * | aggr | <p>The runnable has a mode switch point. The aggregation of ModeSwitchPoint is subject to variability with the purpose to support the conditional existence of mode ports or the variant existence of mode switch points in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: <code>atp.Splitkey=modeSwitchPoint.shortName, modeSwitchPoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</code></p> |
| parameter Access | ParameterAccess | * | aggr | <p>The presence of a ParameterAccess implies that a RunnableEntity needs read only access to a Parameter DataPrototype which may either be local or within a Port Prototype.</p> <p>The aggregation of ParameterAccess is subject to variability with the purpose to support the conditional existence of parameter ports and component local parameters as well as the variant existence of Parameter Access (points) in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: <code>atp.Splitkey=parameterAccess.shortName, parameterAccess.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</code></p> |
| readLocal Variable | VariableAccess | * | aggr | <p>The presence of a readLocalVariable implies that a RunnableEntity needs read access to a VariableData Prototype in the role of implicitInterRunnableVariable or explicitInterRunnableVariable.</p> <p>The aggregation of readLocalVariable is subject to variability with the purpose to support the conditional existence of implicitInterRunnableVariable and explicit InterRunnableVariable or the variant existence of read LocalVariable (points) in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: <code>atp.Splitkey=readLocalVariable.shortName, readLocalVariable.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</code></p> |
| serverCallPoint | ServerCallPoint | * | aggr | <p>The RunnableEntity has a ServerCallPoint. The aggregation of ServerCallPoint is subject to variability with the purpose to support the conditional existence of client server PortPrototypes or the variant existence of server call points in the implementation.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: <code>atp.Splitkey=serverCallPoint.shortName, serverCallPoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</code></p> |





| Class | RunnableEntity | | | |
|----------------------|----------------|------|------|--|
| symbol | CIdentifier | 0..1 | attr | The symbol describing this RunnableEntity's entry point. This is considered the API of the RunnableEntity and is required during the RTE contract phase. |
| waitPoint | WaitPoint | * | aggr | The WaitPoint associated with the RunnableEntity. |
| writtenLocalVariable | VariableAccess | * | aggr | <p>The presence of a writtenLocalVariable implies that a RunnableEntity needs write access to a VariableData Prototype in the role of implicitInterRunnableVariable or explicitInterRunnableVariable.</p> <p>The aggregation of writtenLocalVariable is subject to variability with the purpose to support the conditional existence of implicitInterRunnableVariable and explicitInterRunnableVariable or the variant existence of written LocalVariable (points) in the implementation.</p> <p>Stereotypes: atpSplittable; atpVariation</p> <p>Tags: atp.Splitkey=writtenLocalVariable.shortName, written LocalVariable.variationPoint.shortLabel vh.latestBindingTime=preCompileTime </p> |

Table D.48: RunnableEntity

| Class | SenderReceiverInterface | | | |
|--------------------|--|-------|------|---|
| Package | M2::AUTOSARTemplates::SWComponentTemplate::PortInterface | | | |
| Note | A sender/receiver interface declares a number of data elements to be sent and received. | | | |
| Tags: | atp.recommendedPackage=PortInterfaces | | | |
| Base | ARElement, ARObject, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, CollectableElement, DataInterface, Identifiable, MultilanguageReferrable, PackageableElement, PortInterface, Referrable | | | |
| Aggregated by | ARPackage.element | | | |
| Attribute | Type | Mult. | Kind | Note |
| dataElement | VariableDataPrototype | * | aggr | The data elements of this SenderReceiverInterface. |
| invalidationPolicy | InvalidationPolicy | * | aggr | InvalidationPolicy for a particular dataElement |
| metaDataItemSet | MetaDataAdapterSet | * | aggr | This aggregation defines fixed sets of meta-data items associated with dataElements of the enclosing Sender ReceiverInterface |

Table D.49: SenderReceiverInterface

| Class | SensorActuatorSwComponentType | | | |
|----------------|--|-------|------|---|
| Package | M2::AUTOSARTemplates::SWComponentTemplate::Components | | | |
| Note | The SensorActuatorSwComponentType introduces the possibility to link from the software representation of a sensor/actuator to its hardware description provided by the ECU Resource Template. | | | |
| Tags: | atp.recommendedPackage=SwComponentTypes | | | |
| Base | ARElement, ARObject, AtomicSwComponentType, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable, SwComponentType | | | |
| Aggregated by | ARPackage.element | | | |
| Attribute | Type | Mult. | Kind | Note |
| sensorActuator | HwDescriptionEntity | 0..1 | ref | Reference from the Sensor Actuator Software Component Type to the description of the actual hardware. |

Table D.50: SensorActuatorSwComponentType

| | | | | |
|----------------------|--|--------------|-------------|---|
| Class | SwComponentPrototype | | | |
| Package | M2::AUTOSARTemplates::SWComponentTemplate::Composition | | | |
| Note | Role of a software component within a composition. | | | |
| Base | <i>ARObject, AtpFeature, AtpPrototype, Identifiable, MultilanguageReferrable, Referrable</i> | | | |
| Aggregated by | <i>AtpClassifier.atpFeature, CompositionSwComponentType.component</i> | | | |
| Attribute | Type | Mult. | Kind | Note |
| type | SwComponentType | 0..1 | tref | Type of the instance. Stereotypes: isOfType |

Table D.51: SwComponentPrototype

| | | | | |
|------------------------------|---|--------------|-------------|---|
| Class | SwComponentType (abstract) | | | |
| Package | M2::AUTOSARTemplates::SWComponentTemplate::Components | | | |
| Note | Base class for AUTOSAR software components. | | | |
| Base | <i>ARElement, ARObject, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable</i> | | | |
| Subclasses | <i>AtomicSwComponentType, CompositionSwComponentType, ParameterSwComponentType</i> | | | |
| Aggregated by | <i>ARPackage.element</i> | | | |
| Attribute | Type | Mult. | Kind | Note |
| consistency Needs | ConsistencyNeeds | * | aggr | This represents the collection of ConsistencyNeeds owned by the enclosing SwComponentType. Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=consistencyNeeds.shortName, consistency Needs.variationPoint.shortLabel vh.latestBindingTime=preCompileTime |
| port | PortPrototype | * | aggr | The PortPrototypes through which this SwComponent Type can communicate. The aggregation of PortPrototype is subject to variability with the purpose to support the conditional existence of PortPrototypes. Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=port.shortName, port.variationPoint.short Label vh.latestBindingTime=preCompileTime |
| portGroup | PortGroup | * | aggr | A port group being part of this component. Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=portGroup.shortName, portGroup.variation Point.shortLabel vh.latestBindingTime=preCompileTime |
| swcMapping Constraint | SwComponentMapping Constraints | * | ref | Reference to constraints that are valid for this Sw ComponentType. |
| swComponent Documentation | SwComponent Documentation | 0..1 | aggr | This adds a documentation to the SwComponentType. Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=swComponentDocumentation, sw ComponentDocumentation.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=-10 |
| unitGroup | UnitGroup | * | ref | This allows for the specification of which UnitGroups are relevant in the context of referencing SwComponentType. |

Table D.52: SwComponentType

| | | | | |
|-------------------------|---|--------------|-------------|--|
| Class | SwSystemconstantValueSet | | | |
| Package | M2::AUTOSARTemplates::GenericStructure::VariantHandling | | | |
| Note | This meta-class represents the ability to specify a set of system constant values. Tags: atp.recommendedPackage=SwSystemconstantValueSets | | | |
| Base | ARElement, AROObject, CollectableElement, <i>Identifiable</i> , MultilanguageReferrable, Packageable Element, <i>Referrable</i> | | | |
| Aggregated by | ARPackage.element | | | |
| Attribute | Type | Mult. | Kind | Note |
| sw Systemconstant Value | SwSystemconstValue | * | aggr | This is one particular value of a system constant. |

Table D.53: SwSystemconstantValueSet

| | | | | |
|----------------------------|--|--------------|-------------|--|
| Class | SwcInternalBehavior | | | |
| Package | M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior | | | |
| Note | The SwcInternalBehavior of an AtomicSwComponentType describes the relevant aspects of the software-component with respect to the RTE, i.e. the RunnableEntities and the RTEEvents they respond to. | | | |
| Base | ARObject, AtpClassifier, AtpFeature, AtpStructureElement, <i>Identifiable</i> , <i>InternalBehavior</i> , Multilanguage Referrable, <i>Referrable</i> | | | |
| Aggregated by | AtomicSwComponentType.internalBehavior, AtpClassifier.atpFeature | | | |
| Attribute | Type | Mult. | Kind | Note |
| arTypedPer Instance Memory | VariableDataPrototype | * | aggr | <p>Defines an AUTOSAR typed memory-block that needs to be available for each instance of the SW-component.</p> <p>This is typically only useful if supportsMultipleInstantiation is set to "true" or if the component defines NVRAM access via permanent blocks.</p> <p>The aggregation of arTypedPerInstanceMemory is subject to variability with the purpose to support variability in the software component's implementations. Typically different algorithms in the implementation are requiring different number of memory objects.</p> <p>Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=arTypedPerInstanceMemory.shortName, ar TypedPerInstanceMemory.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p> |
| event | RTEEvent | * | aggr | <p>This is a RTEEvent specified for the particular Swc InternalBehavior.</p> <p>The aggregation of RTEEvent is subject to variability with the purpose to support the conditional existence of RTE events. Note: the number of RTE events might vary due to the conditional existence of PortPrototypes using Data ReceivedEvents or due to different scheduling needs of algorithms.</p> <p>Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=event.shortName, event.variationPoint.short Label vh.latestBindingTime=preCompileTime</p> |





| Class | SwcInternalBehavior | | | |
|-----------------------------------|----------------------------------|---|------|---|
| exclusiveArea Policy | SwcExclusiveArea Policy | * | aggr | <p>Options how to generate the ExclusiveArea related APIs. When no SwcExclusiveAreaPolicy is specified for an ExclusiveArea the default values apply.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: <code>atp.Splitkey=exclusiveAreaPolicy, exclusiveArea Policy.variationPoint.shortLabel</code> <code>vh.latestBindingTime=preCompileTime</code></p> |
| explicitInter Runnable Variable | VariableDataPrototype | * | aggr | <p>Implement state message semantics for establishing communication among runnables of the same component. The aggregation of explicitInterRunnable Variable is subject to variability with the purpose to support variability in the software components implementations. Typically different algorithms in the implementation are requiring different number of memory objects.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: <code>atp.Splitkey=explicitInterRunnableVariable.shortName, explicitInterRunnableVariable.variationPoint.shortLabel</code> <code>vh.latestBindingTime=preCompileTime</code></p> |
| implicitInter Runnable Variable | VariableDataPrototype | * | aggr | <p>Implement state message semantics for establishing communication among runnables of the same component. The aggregation of implicitInterRunnable Variable is subject to variability with the purpose to support variability in the software components implementations. Typically different algorithms in the implementation are requiring different number of memory objects.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: <code>atp.Splitkey=implicitInterRunnableVariable.shortName, implicitInterRunnableVariable.variationPoint.shortLabel</code> <code>vh.latestBindingTime=preCompileTime</code></p> |
| includedData TypeSet | IncludedDataTypeSet | * | aggr | <p>The includedDataTypeSet is used by a software component for its implementation.</p> <p>Stereotypes: atpSplitable Tags: <code>atp.Splitkey=includedDataTypeSet</code></p> |
| includedMode Declaration GroupSet | IncludedMode DeclarationGroupSet | * | aggr | <p>This aggregation represents the included Mode DeclarationGroups</p> <p>Stereotypes: atpSplitable Tags: <code>atp.Splitkey=includedModeDeclarationGroupSet</code></p> |
| instantiation DataDefProps | InstantiationDataDef Props | * | aggr | <p>The purpose of this is that within the context of a given SwComponentType some data def properties of individual instantiations can be modified. The aggregation of InstantiationDataDefProps is subject to variability with the purpose to support the conditional existence of Port Prototypes and component local memories like "per InstanceParameter" or "arTypedPerInstanceMemory".</p> <p>Stereotypes: atpSplitable; atpVariation Tags: <code>atp.Splitkey=instantiationDataDefProps, instantiationData DefProps.variationPoint.shortLabel</code> <code>vh.latestBindingTime=preCompileTime</code></p> |





| Class | SwcInternalBehavior | | | |
|-----------------------|-------------------------|---|------|--|
| perInstance Memory | PerInstanceMemory | * | aggr | <p>Defines a per-instance memory object needed by this software component. The aggregation of PerInstance Memory is subject to variability with the purpose to support variability in the software components implementations. Typically different algorithms in the implementation are requiring different number of memory objects.</p> <p>Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=perInstanceMemory.shortName, perInstance Memory.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p> |
| perInstance Parameter | ParameterData Prototype | * | aggr | <p>Defines parameter(s) or characteristic value(s) that needs to be available for each instance of the software-component. This is typically only useful if supportsMultipleInstantiation is set to "true". The aggregation of perInstanceParameter is subject to variability with the purpose to support variability in the software components implementations. Typically different algorithms in the implementation are requiring different number of memory objects.</p> <p>Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=perInstanceParameter.shortName, per InstanceParameter.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p> |
| portAPIOption | PortAPIOption | * | aggr | <p>Options for generating the signature of port-related calls from a runnable to the RTE and vice versa. The aggregation of PortPrototypes is subject to variability with the purpose to support the conditional existence of ports.</p> <p>Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=portAPIOption, portAPIOption.variation Point.shortLabel vh.latestBindingTime=preCompileTime</p> |
| runnable | RunnableEntity | * | aggr | <p>This is a RunnableEntity specified for the particular Swc InternalBehavior.</p> <p>The aggregation of RunnableEntity is subject to variability with the purpose to support the conditional existence of RunnableEntities. Note: the number of RunnableEntities might vary due to the conditional existence of Port Prototypes using DataReceivedEvents or due to different scheduling needs of algorithms.</p> <p>Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=runnable.shortName, runnable.variation Point.shortLabel vh.latestBindingTime=preCompileTime</p> |





| Class | SwInternalBehavior | | | |
|---------------------------------|-------------------------|------|------|---|
| service Dependency | SwcService Dependency | * | aggr | <p>Defines the requirements on AUTOSAR Services for a particular item.</p> <p>The aggregation of SwcServiceDependency is subject to variability with the purpose to support the conditional existence of ports as well as the conditional existence of ServiceNeeds.</p> <p>The SwcServiceDependency owned by an SwInternal Behavior can be located in a different physical file in order to support that SwcServiceDependency might be provided in later development steps or even by different expert domain (e.g OBD expert for OBD related Service Needs) tools. Therefore the aggregation is <>atp Splitable>>.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=serviceDependency.shortName, service Dependency.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p> |
| shared Parameter | ParameterData Prototype | * | aggr | <p>Defines parameter(s) or characteristic value(s) shared between SwComponentPrototypes of the same Sw ComponentType. The aggregation of sharedParameter is subject to variability with the purpose to support variability in the software components implementations. Typically different algorithms in the implementation are requiring different number of memory objects.</p> <p>Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=sharedParameter.shortName, shared Parameter.variationPoint.shortLabel vh.latestBindingTime=preCompileTime</p> |
| supports Multiple Instantiation | Boolean | 0..1 | attr | Indicate whether the corresponding software-component can be multiply instantiated on one ECU. In this case the attribute will result in an appropriate component API on programming language level (with or without instance handle). |
| variationPoint Proxy | VariationPointProxy | * | aggr | <p>Proxy of a variation points in the C/C++ implementation.</p> <p>Stereotypes: atpSplitable Tags: atp.Splitkey=variationPointProxy.shortName</p> |

Table D.54: SwInternalBehavior

| Class | System | | | | |
|---------------|---|------|-------|------|------|
| Package | M2::AUTOSARTemplates::SystemTemplate | | | | |
| Note | <p>The top level element of the System Description. The System description defines five major elements: Topology, Software, Communication, Mapping and Mapping Constraints.</p> <p>The System element directly aggregates the elements describing the Software, Mapping and Mapping Constraints; it contains a reference to an ASAM FIBEX description specifying Communication and Topology.</p> <p>Tags: atp.recommendedPackage=Systems</p> | | | | |
| Base | <i>ARElement, ARObject, AtpClassifier, AtpFeature, AtpStructureElement, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable, UploadableDesignElement, UploadablePackageElement</i> | | | | |
| Aggregated by | <i>ARPackage.element, AtpClassifier.atpFeature</i> | | | | |
| Attribute | <table border="1"> <thead> <tr> <th>Type</th> <th>Mult.</th> <th>Kind</th> <th>Note</th> </tr> </thead> </table> | Type | Mult. | Kind | Note |
| Type | Mult. | Kind | Note | | |





| Class | System | | | |
|--------------------------------|--------------------------------|------|------|---|
| clientIdDefinitionSet | ClientIdDefinitionSet | * | ref | Set of Client Identifiers that are used for inter-ECU client-server communication in the System. |
| containerIPduHeaderByteOrder | ByteOrderEnum | 0..1 | attr | Defines the byteOrder of the header in ContainerIPdus. |
| ecuExtractVersion | RevisionLabelString | 0..1 | attr | Version number of the Ecu Extract. |
| fibexElement | FibexElement | * | ref | <p>Reference to ASAM FIBEX elements specifying Communication and Topology.</p> <p>All Fibex Elements used within a System Description shall be referenced from the System Element.</p> <p>atpVariation: In order to describe a product-line, all Fibex Elements can be optional.</p> <p>Stereotypes: atpSplittable; atpVariation</p> <p>Tags: atp.Splitkey=fibexElement.fibexElement, fibex Element.variationPoint.shortLabel vh.latestBindingTime=postBuild </p> |
| interpolationRoutineMappingSet | InterpolationRoutineMappingSet | * | ref | This reference identifies the InterpolationRoutineMapping Sets that are relevant in the context of the enclosing System. |
| j1939SharedAddressCluster | J1939SharedAddressCluster | * | aggr | <p>Collection of J1939Clusters that share a common address space for the routing of messages.</p> <p>Stereotypes: atpSplittable; atpVariation</p> <p>Tags: atp.Splitkey=j1939SharedAddressCluster.shortName, j1939SharedAddressCluster.variationPoint.shortLabel vh.latestBindingTime=postBuild </p> |
| mapping | SystemMapping | * | aggr | <p>Aggregation of all mapping aspects (mapping of SW components to ECUs, mapping of data elements to signals, and mapping constraints).</p> <p>In order to support OEM / Tier 1 interaction and shared development for one common System this aggregation is atpSplittable and atpVariation. The content of System Mapping can be provided by several parties using different names for the SystemMapping.</p> <p>This element is not required when the System description is used for a network-only use-case.</p> <p>Stereotypes: atpSplittable; atpVariation</p> <p>Tags: atp.Splitkey=mapping.shortName, mapping.variation Point.shortLabel vh.latestBindingTime=postBuild </p> |
| pncVectorLength | PositiveInteger | 0..1 | attr | Length of the partial networking request release information vector (in bytes). |
| pncVectorOffset | PositiveInteger | 0..1 | attr | Absolute offset (with respect to the NM-PDU) of the partial networking request release information vector that is defined in bytes as an index starting with 0. |





| Class | System | | | |
|-------------------------|----------------------------|------|------|---|
| rootSoftwareComposition | RootSwCompositionPrototype | 0..1 | aggr | <p>Aggregation of the root software composition, containing all software components in the System in a hierarchical structure. This element is not required when the System description is used for a network-only use-case.</p> <p>atpVariation: The RootSwCompositionPrototype can vary.</p> <p>Stereotypes: atpSplittable; atpVariation</p> <p>Tags: atp.Splitkey=rootSoftwareComposition.shortName, rootSoftwareComposition.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime</p> |
| swCluster | CpSoftwareCluster | * | ref | <p>CP Software Clusters of this System</p> <p>Stereotypes: atpSplittable; atpVariation</p> <p>Tags: atp.Splitkey=swCluster.cpSoftwareCluster, swCluster.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime</p> |
| systemDocumentation | Chapter | * | aggr | <p>Possibility to provide additional documentation while defining the System. The System documentation can be composed of several chapters.</p> <p>Stereotypes: atpSplittable; atpVariation</p> <p>Tags: atp.Splitkey=systemDocumentation.shortName, systemDocumentation.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=-10</p> |
| systemVersion | RevisionLabelString | 0..1 | attr | Version number of the System Description. |

Table D.55: System

| | | | | |
|----------------------|--|-------|------|---------------------------------|
| Class | TDEventBsw (abstract) | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescriptionEvents::TDEventBsw | | | |
| Note | This is used to describe timing events related to BSW modules. | | | |
| Base | ARObject, Identifiable, MultilanguageReferrable, Referrable, TimingDescription, TimingDescriptionEvent | | | |
| Subclasses | TDEventBswModeDeclaration, TDEventBswModule | | | |
| Aggregated by | TimingExtension.timingDescription | | | |
| Attribute | Type | Mult. | Kind | Note |
| bswModuleDescription | BswModuleDescription | 0..1 | ref | The scope of this timing event. |

Table D.56: TDEventBsw

| | | | | |
|---------------|--|-------|------|------|
| Class | TDEventVfbPortGroup | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescriptionEvents::TDEventVfb | | | |
| Note | A TimingDescriptionEvent occuring on a heterogeneous group of TDEventVfbPort. | | | |
| Tags: | atp.Status=draft | | | |
| Base | ARObject, Identifiable, MultilanguageReferrable, Referrable, TDEventVfb, TimingDescription, TimingDescriptionEvent | | | |
| Aggregated by | TimingExtension.timingDescription | | | |
| Attribute | Type | Mult. | Kind | Note |





| Class | TDEventVfbPortGroup | | | |
|----------------|---------------------|---|-----|--|
| tdEventVfbPort | TDEventVfbPort | * | ref | Group of PortPrototype s on which the timing events shall apply Tags: atp.Status=draft |

Table D.57: TDEventVfbPortGroup

| Class | TDLETZoneClock | | | |
|---------------|---|-------|------|---|
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingClock | | | |
| Note | Represents a LET Zone Clock Tags: atp.Status=draft | | | |
| Base | ARObject, Identifiable , MultilanguageReferrable , Referrable , TimingClock | | | |
| Aggregated by | TimingExtension.timingClock | | | |
| Attribute | Type | Mult. | Kind | Note |
| accuracyExt | MultidimensionalTime | 0..1 | aggr | Constrains an upper-bounds for any TimingClockSyncAccuracy.accuracy plus the accuracyExt of the TDLETZoneClock , that is referenced by the TimingClockSyncAccuracy . Tags: atp.Status=draft |
| accuracyInt | MultidimensionalTime | 0..1 | aggr | Constrains an upper-bounds for any combination of two accuracy s of referenced TimingClockSyncAccuracy s plus the accuracyExt s of the TDLETZoneClock s, that are referenced by those TimingClockSyncAccuracy s. Tags: atp.Status=draft |

Table D.58: TDLETZoneClock

| Class | TimingClock (abstract) | | | |
|------------------|---|-------|------|--|
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingClock | | | |
| Note | Provides an abstract model of a timing extensions time base. Tags: atp.Status=draft | | | |
| Base | ARObject, Identifiable , MultilanguageReferrable , Referrable | | | |
| Subclasses | TDLETZoneClock | | | |
| Aggregated by | TimingExtension.timingClock | | | |
| Attribute | Type | Mult. | Kind | Note |
| platformTimeBase | GlobalTimeDomain | 0..1 | ref | Refers to a physical time base reference on the respective platform level Stereotypes: atpSplittable; atpVariation Tags: atp.Splitkey=platformTimeBase.globalTimeDomain, platformTimeBase.variationPoint.shortLabel vh.latestBindingTime=postBuild |

Table D.59: TimingClock

| | | | | |
|----------------------|---|--------------|-------------|--|
| Class | TimingClockSyncAccuracy | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingClock | | | |
| Note | Describes the synchronization accuracy between exactly two TDClocks. Tags: atp.Status=draft | | | |
| Base | ARObject, Identifiable , MultilanguageReferrable , Referrable | | | |
| Aggregated by | TimingExtension.timingClockSyncAccuracy | | | |
| Attribute | Type | Mult. | Kind | Note |
| accuracy | MultidimensionalTime | 0..1 | aggr | Synchronization accuracy, treated as zero if not given. Tags: atp.Status=draft |
| lower | TimingClock | 0..1 | ref | Reference a lower level TimingClock , which is derived from a higher level TimingClock . |
| upper | TimingClock | 0..1 | ref | Reference to a higher level TimingClock , from which a lower level TimingClock is derived. |

Table D.60: TimingClockSyncAccuracy

| | | | | |
|----------------------|--|--------------|-------------|---|
| Class | TimingConstraint (abstract) | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint | | | |
| Note | The abstract parent class of different timing constraints supported by the Timing extension. A concrete timing constraint is used to bound the timing behavior of the model elements in its scope. | | | |
| Base | ARObject, Identifiable , MultilanguageReferrable , Referrable , Traceable | | | |
| Subclasses | AgeConstraint , EventTriggeringConstraint , ExecutionOrderConstraint , ExecutionTimeConstraint , LatencyTimingConstraint , OffsetTimingConstraint , SynchronizationPointConstraint , SynchronizationTimingConstraint | | | |
| Aggregated by | TimingExtension.timingGuarantee , TimingExtension.timingRequirement | | | |
| Attribute | Type | Mult. | Kind | Note |
| timingCondition | TimingCondition | 0..1 | ref | A timing condition the timing constraint depends on. In other words it specifies the condition the timing constraint holds. |

Table D.61: TimingConstraint

| | | | | |
|----------------------|---|--------------|-------------|-------------|
| Class | TimingDescription (abstract) | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription | | | |
| Note | The abstract parent class of the model elements that are used to define the scope of a timing constraint. | | | |
| Base | ARObject, Identifiable , MultilanguageReferrable , Referrable | | | |
| Subclasses | TimingDescriptionEvent , TimingDescriptionEventChain | | | |
| Aggregated by | TimingExtension.timingDescription | | | |
| Attribute | Type | Mult. | Kind | Note |
| - | - | - | - | - |

Table D.62: TimingDescription

| | | | | |
|-----------------------|--|--------------|-------------|--|
| Class | <i>TimingDescriptionEvent</i> (abstract) | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription | | | |
| Note | <p>A timing event is the abstract representation of a specific system behavior -- that can be observed at runtime -- in the AUTOSAR specification. Timing events are used to define the scope for timing constraints. Depending on the specific scope, the view on the system, and the level of abstraction different types of events are defined.</p> <p>In order to avoid confusion with existing event descriptions in the AUTOSAR templates the timing specific event types use the prefix TD.</p> | | | |
| Base | <i>ARObject</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i> , <i>TimingDescription</i> | | | |
| Subclasses | <i>TDEventBsw</i> , <i>TDEventBswInternalBehavior</i> , <i>TDEventCom</i> , <i>TDEventComplex</i> , <i>TDEventSwc</i> , <i>TDEventVfb</i> | | | |
| Aggregated by | <i>TimingExtension.timingDescription</i> | | | |
| Attribute | Type | Mult. | Kind | Note |
| clockReference | <i>TimingClock</i> | 0..1 | ref | Optional reference to a clock that holds the time base for an TD event. Tags: atp.Status=draft |
| occurrence Expression | <i>TDEventOccurrence Expression</i> | 0..1 | aggr | The occurrence expression for this event. |

 Table D.63: **TimingDescriptionEvent**

| | | | | |
|----------------------|---|--------------|-------------|---|
| Class | <i>TimingEvent</i> | | | |
| Package | M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents | | | |
| Note | This event is used to start RunnableEntities that shall be executed periodically. | | | |
| Base | <i>ARObject</i> , <i>AbstractEvent</i> , <i>AtpClassifier</i> , <i>AtpFeature</i> , <i>AtpStructureElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>RTEEvent</i> , <i>Referrable</i> | | | |
| Aggregated by | <i>AtpClassifier.atpFeature</i> , <i>SwcInternalBehavior.event</i> | | | |
| Attribute | Type | Mult. | Kind | Note |
| offset | <i>TimeValue</i> | 0..1 | attr | The value makes an assumption about the time offset of the first activation of the RunnableEntity triggered by the mapped TimingEvent relative to the periodic activation of the time base of this TimingEvent. Unit: second. |
| period | <i>TimeValue</i> | 0..1 | attr | Period of timing event in seconds. The value of this attribute shall be greater than zero. |

 Table D.64: **TimingEvent**

| | | | | |
|----------------------|---|--------------|-------------|-------------|
| Class | <i>TimingExtension</i> (abstract) | | | |
| Package | M2::AUTOSARTemplates::CommonStructure::Timing::TimingExtensions | | | |
| Note | <p>The abstract parent class of the different template specific timing extensions.</p> <p>Depending on the specific timing extension the timing descriptions and timing constraints, that can be used to specify the timing behavior, are restricted.</p> | | | |
| Base | <i>ARElement</i> , <i>ARObject</i> , <i>CollectableElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>Referrable</i> | | | |
| Subclasses | <i>BswCompositionTiming</i> , <i>BswModuleTiming</i> , <i>EcuTiming</i> , <i>SwcTiming</i> , <i>SystemTiming</i> , <i>VfbTiming</i> | | | |
| Aggregated by | <i>ARPackage.element</i> | | | |
| Attribute | Type | Mult. | Kind | Note |





| Class | TimingExtension (abstract) | | | |
|--------------------------|----------------------------|---|------|--|
| timingClock | TimingClock | * | aggr | <p>A list of abstract model Clocks.</p> <p>Stereotypes: atpSplitable; atpVariation</p> <p>Tags: atp.Splitkey=timingClock.shortName, timing Clock.variationPoint.shortLabel atp.Status=draft vh.latestBindingTime=postBuild </p> |
| timingClock SyncAccuracy | TimingClockSyncAccuracy | * | aggr | <p>A list of accuracies - which may be used to specify synchronizations from one model clock to another model clock.</p> <p>Stereotypes: atpSplitable; atpVariation</p> <p>Tags: atp.Splitkey=timingClockSyncAccuracy.shortName, timing ClockSyncAccuracy.variationPoint.shortLabel atp.Status=draft vh.latestBindingTime=postBuild </p> |
| timingCondition | TimingCondition | * | aggr | <p>The timing condition specifies a specific condition.</p> <p>Stereotypes: atpSplitable; atpVariation</p> <p>Tags: atp.Splitkey=timingCondition.shortName, timing Condition.variationPoint.shortLabel vh.latestBindingTime=postBuild </p> |
| timing Description | TimingDescription | * | aggr | <p>The timing descriptions that belong to a specific timing specification.</p> <p>In order to support different timing description variants within a timing specification, the aggregation is marked with the stereotype "atpVariation".</p> <p>Stereotypes: atpSplitable; atpVariation</p> <p>Tags: atp.Splitkey=timingDescription.shortName, timing Description.variationPoint.shortLabel vh.latestBindingTime=postBuild </p> |
| timing Guarantee | TimingConstraint | * | aggr | <p>The timing constraints that belong to a specific timing specification in the role of a timing guarantee.</p> <p>In order to support different timing constraint variants within a timing specification, the aggregation is marked with the stereotype "atpVariation".</p> <p>Stereotypes: atpSplitable; atpVariation</p> <p>Tags: atp.Splitkey=timingGuarantee.shortName, timing Guarantee.variationPoint.shortLabel vh.latestBindingTime=postBuild </p> |
| timing Requirement | TimingConstraint | * | aggr | <p>The timing constraints that belong to a specific timing specification in the role of a timing requirement.</p> <p>In order to support different timing constraint variants within a timing specification, the aggregation is marked with the stereotype "atpVariation".</p> <p>Stereotypes: atpSplitable; atpVariation</p> <p>Tags: atp.Splitkey=timingRequirement.shortName, timing Requirement.variationPoint.shortLabel vh.latestBindingTime=postBuild </p> |





| Class | TimingExtension (abstract) | | | |
|----------------|----------------------------|------|------|--|
| timingResource | TimingExtension Resource | 0..1 | aggr | <p>The timing resource contains all instance references referred from within a timing condition formula of a timing view.</p> <p>Stereotypes: atpSplittable</p> <p>Tags: atp.Splitkey=timingResource.shortName</p> |

Table D.65: TimingExtension

| Class | Trigger | | | |
|---------------|--|-------|------|--|
| Package | M2::AUTOSARTemplates::CommonStructure::TriggerDeclaration | | | |
| Note | A trigger which is provided (i.e. released) or required (i.e. used to activate something) in the given context. | | | |
| Base | ARObject, AtpClassifier, AtpFeature, AtpStructureElement, <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i> | | | |
| Aggregated by | AtpClassifier.atpFeature, BswModuleDescription.releasedTrigger, BswModuleDescription.requiredTrigger, ServiceInterface.trigger, TriggerInterface.trigger | | | |
| Attribute | Type | Mult. | Kind | Note |
| swImplPolicy | SwImplPolicyEnum | 0..1 | attr | This attribute, when set to value queued, allows for a queued processing of Triggers. |
| triggerPeriod | MultidimensionalTime | 0..1 | aggr | Optional definition of a period in case of a periodically (time or angle) driven external trigger. |

Table D.66: Trigger

| Class | VariableAccess | | | |
|-------------------|--|-------|------|--|
| Package | M2::AUTOSARTemplates::SWComponentTemplate::SwInternalBehavior::DataElements | | | |
| Note | The presence of a VariableAccess implies that a RunnableEntity needs access to a VariableData Prototype. The kind of access is specified by the role in which the class is used. | | | |
| Base | ARObject, AbstractAccessPoint, AtpClassifier, AtpFeature, AtpStructureElement, <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i> | | | |
| Aggregated by | AtpClassifier.atpFeature, ReceiverComSpec.replaceWith, RunnableEntity.dataReadAccess, RunnableEntity.dataReceivePointByArgument, RunnableEntity.dataReceivePointByValue, RunnableEntity.dataSendPoint, RunnableEntity.dataWriteAccess, RunnableEntity.readLocalVariable, RunnableEntity.writtenLocalVariable | | | |
| Attribute | Type | Mult. | Kind | Note |
| accessed Variable | AutosarVariableRef | 0..1 | aggr | This denotes the accessed variable. |
| scope | VariableAccessScope Enum | 0..1 | attr | This attribute allows for constraining the scope of the corresponding communication. For example, it is possible to express whether the communication is intended to cross the boundary of an ECU or whether it is intended not to cross the boundary of a single partition. |

Table D.67: VariableAccess

| Class | VariableDataPrototype | | | |
|---------|---|--|--|--|
| Package | M2::AUTOSARTemplates::SWComponentTemplate::Datatype::DataPrototypes | | | |
| Note | A VariableDataPrototype represents a formalized generic piece of information that is typically mutable by the application software layer. VariableDataPrototype is used in various contexts and the specific context gives the otherwise generic VariableDataPrototype a dedicated semantics. | | | |





| Class | VariableDataPrototype | | | |
|----------------------|--|-------|------|---|
| Base | <i>ARObject, AtpFeature, AtpPrototype, AutosarDataPrototype, DataPrototype, Identifiable, MultilanguageReferrable, Referrable</i> | | | |
| Aggregated by | ApplicationInterface.indication, <i>AtpClassifier.atpFeature, BswInternalBehavior.arTypedPerInstanceMemory, BswModuleDescription.providedData, BswModuleDescription.requiredData, BulkNvDataDescriptor.bulkNvBlock, InternalBehavior.staticMemory, NvBlockDescriptor.ramBlock, NvDataInterface.nvData, SenderReceiverInterface.dataElement, ServiceInterface.event, SwcInternalBehavior.arTypedPerInstanceMemory, SwcInternalBehavior.explicitInterRunnableVariable, SwcInternalBehavior.implicitInterRunnableVariable</i> | | | |
| Attribute | Type | Mult. | Kind | Note |
| initValue | ValueSpecification | 0..1 | aggr | Specifies initial value(s) of the VariableDataPrototype |

Table D.68: VariableDataPrototype

E Splitable Elements in the Scope of this Document

This chapter contains a table of all model elements stereotyped `<<atpSplitable>>` in the scope of this document.

Each entry in the following table consists of the identification of the specific model element itself and the applicable value of the tagged value `atp.Splitkey`.

For more information about the concept of splitable model elements and how these shall be treated please refer to [8].

| <i>Name of splitable element</i> | <i>Splitkey</i> |
|---|---|
| <code>TDCpSoftwareClusterMappingSet.tdCpSoftwareClusterResourceToTdMapping</code> | <code>tdCpSoftwareClusterResourceToTdMapping.shortName, tdCpSoftwareClusterResourceToTdMapping.variationPoint.shortLabel</code> |
| <code>TDCpSoftwareClusterMappingSet.tdCpSoftwareClusterToTdMapping</code> | <code>tdCpSoftwareClusterToTdMapping.shortName, tdCpSoftwareClusterToTdMapping.variationPoint.shortLabel</code> |
| <code>TimingClock.platformTimeBase</code> | <code>platformTimeBase.globalTimeDomain, platformTimeBase.variationPoint.shortLabel</code> |
| <code>TimingExtension.timingClock</code> | <code>timingClock.shortName, timingClock.variationPoint.shortLabel</code> |
| <code>TimingExtension.timingClockSyncAccuracy</code> | <code>timingClockSyncAccuracy.shortName, timingClockSyncAccuracy.variationPoint.shortLabel</code> |
| <code>TimingExtension.timingCondition</code> | <code>timingCondition.shortName, timingCondition.variationPoint.shortLabel</code> |
| <code>TimingExtension.timingDescription</code> | <code>timingDescription.shortName, timingDescription.variationPoint.shortLabel</code> |
| <code>TimingExtension.timingGuarantee</code> | <code>timingGuarantee.shortName, timingGuarantee.variationPoint.shortLabel</code> |
| <code>TimingExtension.timingRequirement</code> | <code>timingRequirement.shortName, timingRequirement.variationPoint.shortLabel</code> |
| <code>TimingExtension.timingResource</code> | <code>timingResource.shortName</code> |
| <code>TimingExtensionResource.timingArgument</code> | <code>timingArgument.shortName, timingArgument.variationPoint.shortLabel</code> |
| <code>TimingExtensionResource.timingMode</code> | <code>timingMode.shortName, timingMode.variationPoint.shortLabel</code> |
| <code>TimingExtensionResource.timingVariable</code> | <code>timingVariable.shortName, timingVariable.variationPoint.shortLabel</code> |

Table E.1: Usage of splitable elements

F Variation Points in the Scope of this Document

This chapter contains a table of all model elements stereotyped `<<atpVariation>>` in the scope of this document.

Each entry in the following table consists of the identification of the model element itself and the applicable value of the tagged value `vh.latestBindingTime`.

For more information about the concept of variation points and how model elements that contain variation points shall be treated please refer to [8].

| <i>Variation Point</i> | <i>Latest Binding Time</i> |
|---|-----------------------------------|
| <code>TDCpSoftwareClusterMappingSet.tdCpSoftwareClusterResourceToTdMapping</code> | postBuild |
| <code>TDCpSoftwareClusterMappingSet.tdCpSoftwareClusterToTdMapping</code> | postBuild |
| <code>TimingClock.platformTimeBase</code> | postBuild |
| <code>TimingExtension.timingClock</code> | postBuild |
| <code>TimingExtension.timingClockSyncAccuracy</code> | postBuild |
| <code>TimingExtension.timingCondition</code> | postBuild |
| <code>TimingExtension.timingDescription</code> | postBuild |
| <code>TimingExtension.timingGuarantee</code> | postBuild |
| <code>TimingExtension.timingRequirement</code> | postBuild |
| <code>TimingExtensionResource.timingArgument</code> | postBuild |
| <code>TimingExtensionResource.timingMode</code> | postBuild |
| <code>TimingExtensionResource.timingVariable</code> | postBuild |

Table F.1: Usage of variation points

G Change History

Please note that the lists in this chapter also include constraints and specification items that have been removed from the specification in a later version. These constraints and specification items do not appear as hyperlinks in the document.

G.1 Change History of this document according to AUTOSAR Release R4.3.0

G.1.1 Added Specification Items in Release R4.3.0

| Number | Heading |
|------------------------|--|
| [TPS_TIMEX_- 00049] | Purpose of TimingCondition |
| [TPS_TIMEX_- 00050] | Purpose of TimingConditionFormula |
| [TPS_TIMEX_- 00051] | Purpose of TimingExtensionResource |
| [TPS_TIMEX_- 00052] | Purpose of TDEventFrameEthernet |

Table G.1: Added Traceables in 4.3.0

G.1.2 Changed Specification Items in Release R4.3.0

none

G.1.3 Added Constraints in Release R4.3.0

| Number | Heading |
|---------------|--|
| [constr_4551] | Use only Numericals in TDEventOccurrenceExpression |
| [constr_4552] | Restricted usage of AutosarVariableInstance for Content Filter |

Table G.2: Added Constraints in R4.3.0

G.1.4 Changed Constraints in Release R4.3.0

| Number | Heading |
|---------------|--|
| [constr_4501] | Application rule for the occurrence expression |

Table G.3: Added Constraints in R4.3.0

G.1.5 Deleted Constraints in Release R4.3.0

none

G.2 Change History of this document according to AUTOSAR Release R4.3.1

G.2.1 Added Specification Items in Release R4.3.1

none

G.2.2 Changed Specification Items in Release R4.3.1

none

G.2.3 Added Constraints in Release R4.3.1

none

G.2.4 Changed Constraints in Release R4.3.1

none

G.2.5 Deleted Constraints in Release R4.3.1

none

G.3 Change History of this document according to AUTOSAR Release R4.4.0

G.3.1 Added Specification Items in Release R4.4.0

| Number | Heading |
|--------------------|---|
| [TPS_TIMEX_-00053] | Purpose of BswCompositionTiming |
| [TPS_TIMEX_-00054] | Purpose of SynchronizationPointConstraint |
| [TPS_TIMEX_-00055] | Purpose of letInterval |
| [TPS_TIMEX_-00056] | Attribute category used in context of TimingDescriptionEvent and LET |
| [TPS_TIMEX_-00057] | Attribute category used in context of TimingDescriptionEventChain and LET |

Table G.4: Added Traceables in 4.4.0

G.3.2 Changed Specification Items in Release R4.4.0

none

G.3.3 Added Constraints in Release R4.4.0

| Number | Heading |
|---------------|---|
| [constr_4553] | Usage of optional attribute <code>ExecutionOrderConstraint.ignoreOrderAllowed</code> |
| [constr_4554] | Usage of optional directed association/reference <code>letInterval</code> |
| [constr_4555] | Usage of the category value <code>LET_RELEASE</code> in <code>TimingDescriptionEvent</code> |
| [constr_4556] | Usage of the category value <code>LET_TERMINATE</code> in <code>TimingDescriptionEvent</code> |
| [constr_4557] | Usage of the category value <code>LET_INTERVAL</code> in <code>TimingDescriptionEventChain</code> |
| [constr_4558] | Applicability of LET semantics |
| [constr_4559] | <code>category</code> of <code>TimingDescriptionEvent</code> shall not be extended |
| [constr_4560] | <code>category</code> of <code>TimingDescriptionEventChain</code> shall not be extended |

Table G.5: Added Constraints in R4.4.0

G.3.4 Changed Constraints in Release R4.4.0

none

G.3.5 Deleted Constraints in Release R4.4.0

| Number | Heading |
|---------------|---|
| [constr_4535] | An <code>ExecutionOrderConstraint</code> needs to be consistent regarding effective modes |

Table G.6: Deleted Constraints in R4.4.0

G.4 Change History of this document according to AUTOSAR Release R19-11

G.4.1 Added Specification Items in Release 19-11

none

G.4.2 Changed Specification Items in Release 19-11

none

G.4.3 Added Constraints in Release 19-11

none

G.4.4 Changed Constraints in Release 19-11

none

G.4.5 Deleted Constraints in Release 19-11

none

G.5 Change History of this document according to AUTOSAR Release R20-11**G.5.1 Added Specification Items in R20-11**

| Number | Heading |
|--------------------|---|
| [TPS_TIMEX_-00066] | Purpose of TDCpSoftwareClusterMappingSet |
| [TPS_TIMEX_-00067] | Purpose of TDCpSoftwareClusterMapping |
| [TPS_TIMEX_-00068] | Purpose of TDCpSoftwareClusterResourceMapping |

Table G.7: Added Traceables in R20-11

G.5.2 Changed Specification Items in R20-11

none

G.5.3 Deleted Specification Items in R20-11

none

G.5.4 Added Constraints in R20-11

| Number | Heading |
|---------------|--|
| [constr_4561] | Usage of the category value DISPATCH_ENTRY_POINT in TimingDescriptionEvent |

| | |
|---------------|--|
| [constr_4562] | Usage of the category value <code>DISPATCH_EXIT_POINT</code> in <code>TimingDescriptionEvent</code> |
| [constr_4563] | <code>TDCpSoftwareClusterMapping</code> shall reference only dispatcher or LET interval |
| [constr_4564] | <code>TDCpSoftwareClusterResourceMapping</code> shall reference only dispatcher or LET interval |
| [constr_4565] | Consistency of <code>TDCpSoftwareClusterMapping</code> and <code>TDCpSoftwareClusterResourceMapping</code> |
| [constr_4566] | <code>SystemTiming</code> describing timing of software clusters and category of <code>System</code> |
| [constr_4567] | Reference provider of <code>TDCpSoftwareClusterMapping</code> |
| [constr_4568] | Reference requestor of <code>TDCpSoftwareClusterMapping</code> |

Table G.8: Added Constraints in R20-11

G.5.5 Changed Constraints in R20-11

none

G.5.6 Deleted Constraints in R20-11

none

G.6 Change History of this document according to AUTOSAR Release R21-11

G.6.1 Added Specification Items in R21-11

| Number | Heading |
|-------------------|---|
| [TPS_TIMEX_00001] | Purpose of <code>TimingDescriptionEvent</code> |
| [TPS_TIMEX_00002] | Purpose of <code>TimingDescriptionEventChain</code> |
| [TPS_TIMEX_00003] | <code>EventTriggeringConstraint</code> specifies occurrence behavior respectively model |
| [TPS_TIMEX_00004] | <code>LatencyTimingConstraint</code> specifies latency constraints |
| [TPS_TIMEX_00005] | <code>AgeConstraint</code> to specify age constraints |
| [TPS_TIMEX_00006] | <code>SynchronizationTimingConstraint</code> specifies synchronicity constraints |
| [TPS_TIMEX_00007] | <code>ExecutionOrderConstraint</code> specifies sequence of executing executable entities |
| [TPS_TIMEX_00008] | <code>ExecutionTimeConstraint</code> to specify execution time constraints |
| [TPS_TIMEX_00009] | Optional use of timing extensions |
| [TPS_TIMEX_00010] | <code>PeriodicEventTriggering</code> specifies periodic occurrences of events |





| Number | Heading |
|-------------------|--|
| [TPS_TIMEX_00011] | SporadicEventTriggering specifies sporadic occurrences of events |
| [TPS_TIMEX_00012] | ConcretePatternEventTriggering specifies concrete pattern of occurrences of events |
| [TPS_TIMEX_00013] | BurstPatternEventTriggering specifies burst of occurrences of events |
| [TPS_TIMEX_00014] | ArbitraryEventTriggering specifies arbitrary occurrences of an event |
| [TPS_TIMEX_00015] | OffsetTimingConstraint specifies offset between occurrences of events |
| [TPS_TIMEX_00016] | Purpose of TDEventVfb |
| [TPS_TIMEX_00017] | TDEventVariableDataPrototype specifies events observable at sender/receiver ports |
| [TPS_TIMEX_00018] | TDEventOperation specifies events observable at client/server ports. |
| [TPS_TIMEX_00019] | TDEventModeDeclaration specifies events observable at mode ports. |
| [TPS_TIMEX_00020] | TDEventSwcInternalBehavior specifies observable events of runnable entities |
| [TPS_TIMEX_00021] | Purpose of TDEventCom |
| [TPS_TIMEX_00022] | TDEventISignal specifies events related to the exchange of I-Signals |
| [TPS_TIMEX_00023] | TDEventIPdu specifies events related to the exchange of I-PDUs |
| [TPS_TIMEX_00024] | TDEventFrame specifies events related to the exchange of network frames |
| [TPS_TIMEX_00025] | TDEventFrClusterCycleStart specifies the event related to the start of a FlexRay communication cycle |
| [TPS_TIMEX_00026] | TDEventTTCANCycleStart specifies the event related to the start of a TTCAN communication cycle |
| [TPS_TIMEX_00027] | Purpose of TDEventComplex |
| [TPS_TIMEX_00028] | TDEventBswInternalBehavior specifies observable events of BSW module entities |
| [TPS_TIMEX_00029] | Purpose of TDEventBsw |
| [TPS_TIMEX_00030] | TDEventBswModule specifies observable events when basic software entries are called |
| [TPS_TIMEX_00031] | TDEventBswModeDeclaration specifies observable events in case of BSW mode communication |
| [TPS_TIMEX_00032] | Purpose of VfbTiming |
| [TPS_TIMEX_00033] | Purpose of SwcTiming |
| [TPS_TIMEX_00034] | Purpose of SystemTiming |
| [TPS_TIMEX_00035] | Purpose of BswModuleTiming |
| [TPS_TIMEX_00036] | Purpose of EcuTiming |
| [TPS_TIMEX_00037] | TimingConstraint is a Traceable |
| [TPS_TIMEX_00038] | Purpose of EOCExecutableEntityRefAbstract |
| [TPS_TIMEX_00039] | TDEventTrigger specifies events observable at trigger ports |
| [TPS_TIMEX_00040] | Blueprinting VfbTiming |
| [TPS_TIMEX_00041] | Purpose of EOCExecutableEntityRefGroup |
| [TPS_TIMEX_00042] | Purpose of TDEventVfbPort |



△

| Number | Heading |
|-------------------|--|
| [TPS_TIMEX_00043] | Purpose of <code>TDEventVfbReference</code> |
| [TPS_TIMEX_00044] | Purpose of <code>TDEventSwc</code> |
| [TPS_TIMEX_00045] | Purpose of <code>TDEventSwcInternalBehaviorReference</code> |
| [TPS_TIMEX_00046] | Purpose of <code>EOCExecutableEntityRef</code> |
| [TPS_TIMEX_00047] | Purpose of <code>ExecutionOrderConstraintTypeEnum</code> |
| [TPS_TIMEX_00048] | Purpose of <code>EOCEventRef</code> |
| [TPS_TIMEX_00049] | Purpose of <code>TimingCondition</code> |
| [TPS_TIMEX_00050] | Purpose of <code>TimingConditionFormula</code> |
| [TPS_TIMEX_00051] | Purpose of <code>TimingExtensionResource</code> |
| [TPS_TIMEX_00052] | <code>TDEventFrameEthernet</code> specifies events related to the exchange of Ethernet frames |
| [TPS_TIMEX_00053] | Purpose of <code>BswCompositionTiming</code> |
| [TPS_TIMEX_00054] | <code>SynchronizationPointConstraint</code> explicitly specifies a synchronization point between executable entities |
| [TPS_TIMEX_00055] | Purpose of the attribute <code>letInterval</code> |
| [TPS_TIMEX_00056] | Attribute <code>category</code> used in context of <code>TimingDescriptionEvent</code> and LET |
| [TPS_TIMEX_00057] | Attribute <code>category</code> used in context of <code>TimingDescriptionEventChain</code> and LET |
| [TPS_TIMEX_00066] | Purpose of <code>TDCpSoftwareClusterMappingSet</code> |
| [TPS_TIMEX_00067] | Purpose of <code>TDCpSoftwareClusterMapping</code> |
| [TPS_TIMEX_00068] | Purpose of <code>TDCpSoftwareClusterResourceMapping</code> |

Table G.9: Added Specification Items in R21-11

G.6.2 Changed Specification Items in R21-11

none

G.6.3 Deleted Specification Items in R21-11

none

G.6.4 Added Constraints in R21-11

| Number | Heading |
|---------------|--|
| [constr_4500] | Restricted usage of functions |
| [constr_4501] | Application rule for the occurrence expression in <code>TDEventComplex</code> |
| [constr_4502] | Use references only as function operands |
| [constr_4503] | Restricted usage of <code>AutosarOperationArgumentInstance</code> for Content Filter |
| [constr_4504] | Restricted usage of <code>AgeConstraint</code> |
| [constr_4505] | Specifying minimum and maximum number of occurrences |
| [constr_4506] | Specifying minimum inter-arrival time and pattern length |
| [constr_4507] | Specifying pattern length, pattern jitter and pattern period |
| [constr_4508] | <code>TDEventVfb</code> shall reference <code>PortPrototypeBlueprint</code> only in Blueprints |
| [constr_4509] | Only <code>VfbTiming</code> shall be a Blueprint |
| [constr_4510] | Specifying references to <code>RunnableEntity</code> and <code>VariableAccess</code> |
| [constr_4511] | Validity of referencing <code>RunnableEntity</code> |
| [constr_4512] | Validity of referencing <code>VariableAccess</code> |
| [constr_4513] | <code>SynchronizationTimingConstraint</code> shall reference at least two events |
| [constr_4514] | <code>SynchronizationTimingConstraint</code> shall reference at least two event chains |
| [constr_4515] | Specifying stimulus and response in <code>TimingDescriptionEventChain</code> |
| [constr_4516] | Specifying event chain segments |
| [constr_4517] | Referencing no further event chain segments |
| [constr_4518] | Specifying <code>stimulus</code> event and <code>response</code> event of first and last event chain segment |
| [constr_4519] | Specifying <code>patternLength</code> |
| [constr_4520] | Specifying attribute <code>synchronizationConstraintType</code> |
| [constr_4521] | Specifying attribute <code>synchronizationConstraintType</code> |
| [constr_4522] | <code>SynchronizationTimingConstraint</code> shall either reference events or event chains |
| [constr_4523] | Specifying attributes <code>maxCycles</code> and <code>maxSlots</code> |
| [constr_4524] | Referencing <code>TimingDescriptionEvent</code> |
| [constr_4525] | Precedence of successor relationships <code>successor</code> and <code>directSuccessor</code> |
| [constr_4526] | Specifying <code>maxCycles</code> and <code>maxSlots</code> in a Repetitive Execution Order Constraint |
| [constr_4527] | Referencing <code>TimingDescriptionEvent</code> in a Repetitive Execution Order Constraint |
| [constr_4528] | The <code>root EOCExecutableEntityRefGroup</code> shall reference only <code>EOCExecutableEntityRefGroup</code> s |
| [constr_4529] | Number of nested elements referenced by the <code>root EOCExecutableEntityRefGroup</code> |
| [constr_4530] | An <code>EOCExecutableEntityRefGroup</code> representing a cycle shall reference only <code>EOCExecutableEntityRef</code> s respectively <code>EOCEventRefs</code> |





| Number | Heading |
|---------------|--|
| [constr_4531] | Number of nested elements referenced by <code>EOCExecutableEntityRefGroup</code> representing a cycle |
| [constr_4532] | Successor relationship is not self-referencing |
| [constr_4533] | Maximum number of successor relationships |
| [constr_4534] | Maximum number of <code>directSuccessor</code> relationships |
| [constr_4536] | Compatible recurrence of any <code>ExecutableEntity</code> |
| [constr_4537] | References among elements in an <code>ExecutionOrderConstraint</code> |
| [constr_4538] | Hierarchical Execution Order Constraint: <code>EOCExecutableEntityRef</code> , <code>EOCEventRef</code> , and <code>EOCExecutableEntityRefGroup</code> shall be target or source of a successor relationship |
| [constr_4539] | The successor relationships <code>successor</code> and <code>directSuccessor</code> shall not be used |
| [constr_4540] | <code>maxCycles</code> and <code>maxSlots</code> shall not be zero |
| [constr_4541] | <code>EOCExecutableEntityRef</code> shall reference <code>ExecutableEntity</code> in Ordinary Execution Order Constraint |
| [constr_4542] | <code>EOCExecutableEntityRef</code> shall reference <code>ExecutableEntity</code> in Hierarchical Execution Order Constraint |
| [constr_4543] | Maximum value of the parameter <code>minimumInterArrivalTime</code> |
| [constr_4544] | Specifying <code>patternLength</code> , <code>patternJitter</code> and <code>patternPeriod</code> |
| [constr_4545] | Referring either <code>ExecutableEntity</code> s or <code>AbstractEvents</code> |
| [constr_4546] | Setting the attribute <code>isEvent</code> |
| [constr_4547] | Setting the attribute <code>permitMultipleReferencesToEE</code> |
| [constr_4548] | <code>EOCEventRef</code> shall reference <code>AbstractEvent</code> in Ordinary Execution Order Constraint |
| [constr_4549] | <code>EOCEventRef</code> shall reference <code>AbstractEvent</code> in Hierarchical Execution Order Constraint |
| [constr_4550] | A Hierarchical Execution Order Constraint shall have an unambiguous root <code>EOCExecutableEntityRefGroup</code> |
| [constr_4551] | Use only Numericals in <code>TDEventOccurrenceExpression</code> |
| [constr_4552] | Restricted usage of <code>AutosarVariableInstance</code> for Content Filter |
| [constr_4553] | Usage of optional attribute <code>ignoreOrderAllowed</code> |
| [constr_4554] | Usage of optional directed association/reference <code>letInterval</code> |
| [constr_4555] | Usage of the category value <code>LET_RELEASE</code> in <code>TimingDescriptionEvent</code> |
| [constr_4556] | Usage of the category value <code>LET_TERMINATE</code> in <code>TimingDescriptionEvent</code> |
| [constr_4557] | Usage of the category value <code>LET_INTERVAL</code> in <code>TimingDescriptionEventChain</code> |
| [constr_4558] | Applicability of LET semantics |
| [constr_4559] | <code>category</code> of <code>TimingDescriptionEvent</code> shall not be extended |
| [constr_4560] | <code>category</code> of <code>TimingDescriptionEventChain</code> shall not be extended |
| [constr_4561] | Usage of the category value <code>DISPATCH_ENTRY_POINT</code> in <code>TimingDescriptionEvent</code> |
| [constr_4562] | Usage of the category value <code>DISPATCH_EXIT_POINT</code> in <code>TimingDescriptionEvent</code> |



△

| Number | Heading |
|---------------|--|
| [constr_4563] | <code>TDCpSoftwareClusterMapping</code> shall reference only dispatchers or LET intervals |
| [constr_4564] | <code>TDCpSoftwareClusterResourceMapping</code> shall reference only dispatchers or LET intervals |
| [constr_4565] | Consistency of <code>TDCpSoftwareClusterMapping</code> and <code>TDCpSoftwareClusterResourceMapping</code> |
| [constr_4566] | <code>SystemTiming</code> describing timing of software clusters and category of <code>System</code> |
| [constr_4567] | Reference provider of <code>TDCpSoftwareClusterMapping</code> |
| [constr_4568] | Reference requestor of <code>TDCpSoftwareClusterMapping</code> |

Table G.10: Added Constraints in R21-11

G.6.5 Changed Constraints in R21-11

none

G.6.6 Deleted Constraints in R21-11

none

G.7 Change History of this document according to AUTOSAR Release R22-11

G.7.1 Added Specification Items in R22-11

| Number | Heading |
|-------------------|--|
| [TPS_TIMEX_00100] | Optionality of <code>accuracyInt</code> |
| [TPS_TIMEX_00101] | Semantics of <code>accuracyInt</code> |
| [TPS_TIMEX_00102] | Optionality of <code>LatencyTimingConstraint.minimum</code> used in an LET interval |
| [TPS_TIMEX_00103] | Optionality of <code>accuracyExt</code> |
| [TPS_TIMEX_00104] | Semantics of <code>accuracyExt</code> |
| [TPS_TIMEX_00105] | Purpose of <code>TimingClockSyncAccuracy</code> |
| [TPS_TIMEX_00106] | Purpose of <code>TimingClockSyncAccuracy.upper</code> |
| [TPS_TIMEX_00107] | Purpose of <code>TimingClockSyncAccuracy.lower</code> |
| [TPS_TIMEX_00108] | Usage of the attribute <code>accuracy</code> of <code>TimingClockSyncAccuracy</code> |
| [TPS_TIMEX_00109] | Optionality of <code>accuracy</code> |

▽

△

| Number | Heading |
|-------------------|---|
| [TPS_TIMEX_00110] | Standardized <code>categorys</code> of <code>TimingDescriptionEventChain</code> |
| [TPS_TIMEX_00111] | Semantics of <code>TimingDescriptionEventChain.stimulus</code> |
| [TPS_TIMEX_00112] | Representation of an LET interval <code>release</code> |
| [TPS_TIMEX_00113] | Representation of an LET interval <code>terminate</code> |
| [TPS_TIMEX_00114] | Semantics of <code>TimingDescriptionEventChain.response</code> |
| [TPS_TIMEX_00115] | Representation of an SL-LET interval <code>release</code> |
| [TPS_TIMEX_00116] | Representation of an SL-LET interval <code>terminate</code> |
| [TPS_TIMEX_00117] | Representation of the <i>recurrence</i> of an LET interval |
| [TPS_TIMEX_00118] | Usage of <code>TimingClock.platformTimeBase</code> |
| [TPS_TIMEX_00119] | Purpose of <code>TDLETZoneClock</code> |
| [TPS_TIMEX_00120] | Purpose of <code>TDEventSLLET</code> |
| [TPS_TIMEX_00121] | Representation of the <i>duration</i> of an LET interval |
| [TPS_TIMEX_00122] | Application of a <code>PeriodicEventTriggering</code> constraint used to specify the <i>recurrence</i> of an LET interval |
| [TPS_TIMEX_00123] | Purpose of <code>TimingClock</code> |
| [TPS_TIMEX_00124] | Purpose of <code>TDEventSLLETPort</code> |
| [TPS_TIMEX_00125] | Representation of an SL-LET interval in a <code>TimingDescriptionEventChain</code> |
| [TPS_TIMEX_00126] | Representation of an SL-LET interval in a <code>TimingDescriptionEvent</code> |

Table G.11: Added Specification Items in R22-11

G.7.2 Changed Specification Items in R22-11

| Number | Heading |
|-------------------|--|
| [TPS_TIMEX_00056] | Standardized <code>categorys</code> of <code>TimingDescriptionEvent</code> in Classic Platform |

Table G.12: Changed Specification Items in R22-11

G.7.3 Deleted Specification Items in R22-11

none

G.7.4 Added Constraints in R22-11

| Number | Heading |
|---------------|--|
| [constr_6816] | Restricted usage of <code>TimingDescriptionEventChain.isPipeliningPermitted</code> in <code>TimingDescriptionEventChain</code> |
| [constr_6817] | Restricted usage of <code>TimingDescriptionEvent.clockReference</code> |
| [constr_6818] | Existence of <code>EventTriggeringConstraint.event</code> |
| [constr_6819] | Existence of <code>PeriodicEventTriggering.jitter</code> |
| [constr_6820] | Existence of <code>PeriodicEventTriggering.minimumInterArrivalTime</code> |
| [constr_6821] | Existence of <code>PeriodicEventTriggering.period</code> |
| [constr_6822] | Existence of <code>SporadicEventTriggering.maximumInterArrivalTime</code> |
| [constr_6823] | Existence of <code>SporadicEventTriggering.minimumInterArrivalTime</code> |
| [constr_6824] | Existence of <code>ConcretePatternEventTriggering.patternLength</code> |
| [constr_6825] | Existence of <code>BurstPatternEventTriggering.maxNumberOfOccurrences</code> |
| [constr_6826] | Existence of <code>BurstPatternEventTriggering.minimumInterArrivalTime</code> |
| [constr_6827] | Existence of <code>BurstPatternEventTriggering.patternLength</code> |
| [constr_6828] | Existence of <code>ArbitraryEventTriggering.minimumDistance</code> |
| [constr_6829] | Existence of <code>ArbitraryEventTriggering.maximumDistance</code> |
| [constr_6830] | Existence of <code>ConfidenceInterval.lowerBound</code> |
| [constr_6831] | Existence of <code>ConfidenceInterval.propability</code> |
| [constr_6832] | Existence of <code>ConfidenceInterval.upperBound</code> |
| [constr_6833] | Existence of <code>ExecutionOrderConstraint.orderedElement</code> |
| [constr_6834] | Existence of <code>EOCExecutableEntityRefGroup.nestedElement</code> |
| [constr_6835] | Existence of <code>ExecutionTimeConstraint.executionTimeType</code> |
| [constr_6836] | Existence of <code>ExecutionTimeConstraint.executable</code> |
| [constr_6837] | Existence of <code>LatencyTimingConstraint.latencyConstraintType</code> |
| [constr_6838] | Existence of <code>LatencyTimingConstraint.maximum</code> |
| [constr_6839] | Existence of <code>LatencyTimingConstraint.minimum</code> |
| [constr_6840] | Existence of <code>LatencyTimingConstraint.nominal</code> |
| [constr_6841] | Existence of <code>LatencyTimingConstraint.scope</code> |
| [constr_6842] | Existence of <code>OffsetTimingConstraint.maximum</code> |
| [constr_6843] | Existence of <code>OffsetTimingConstraint.minimum</code> |
| [constr_6844] | Existence of <code>OffsetTimingConstraint.source</code> |
| [constr_6845] | Existence of <code>OffsetTimingConstraint.target</code> |
| [constr_6846] | Existence of <code>SynchronizationTimingConstraint.synchronizationConstraintType</code> |
| [constr_6847] | Existence of <code>SynchronizationTimingConstraint.tolerance</code> |
| [constr_6848] | Existence of <code>VfbTiming.component</code> |
| [constr_6849] | Existence of <code>SystemTiming.system</code> |
| [constr_6850] | Existence of <code>BswModuleTiming.behavior</code> |





| Number | Heading |
|---------------|--|
| [constr_6851] | Existence of <code>BswCompositionTiming.implementation</code> |
| [constr_6852] | Existence of <code>EcuTiming.ecuConfiguration</code> |
| [constr_6853] | Existence of <code>ModeInBswInstanceRef.contextModeDeclarationGroupPrototype</code> |
| [constr_6854] | Existence of <code>ModeInBswInstanceRef.targetModeDeclaration</code> |
| [constr_6855] | Existence of <code>ModeInSwcInstanceRef.contextModeDeclarationGroupPrototype</code> |
| [constr_6856] | Existence of <code>ModeInSwcInstanceRef.contextPort</code> |
| [constr_6857] | Existence of <code>ModeInSwcInstanceRef.targetModeDeclaration</code> |
| [constr_6858] | Existence of <code>TDEventBswInternalBehavior.tdEventBswInternalBehaviorType</code> |
| [constr_6859] | Existence of <code>TDEventBswInternalBehavior.bswModuleEntity</code> |
| [constr_6860] | Existence of <code>TDEventBswModule.tdEventBswModuleType</code> |
| [constr_6861] | Existence of <code>TDEventBswModule.bswModuleEntry</code> |
| [constr_6862] | Existence of <code>TDEventBswModeDeclaration.tdEventBswModeDeclarationType</code> |
| [constr_6863] | Existence of <code>TDEventBswModeDeclaration.modeDeclaration</code> |
| [constr_6864] | Existence of <code>TDEventISignal.tdEventType</code> |
| [constr_6865] | Existence of <code>TDEventISignal.iSignal</code> |
| [constr_6866] | Existence of <code>TDEventISignal.physicalChannel</code> |
| [constr_6867] | Existence of <code>TDEventIPdu.tdEventType</code> |
| [constr_6868] | Existence of <code>TDEventIPdu.iPdu</code> |
| [constr_6869] | Existence of <code>TDEventIPdu.physicalChannel</code> |
| [constr_6870] | Existence of <code>TDEventFrame.tdEventType</code> |
| [constr_6871] | Existence of <code>TDEventFrame.frame</code> |
| [constr_6872] | Existence of <code>TDEventFrame.physicalChannel</code> |
| [constr_6873] | Existence of <code>TDEventFrameEthernet.tdEventType</code> |
| [constr_6874] | Existence of <code>TDHeaderIdRange.maxHeaderId</code> |
| [constr_6875] | Existence of <code>TDHeaderIdRange.minHeaderId</code> |
| [constr_6876] | Existence of <code>TDEventCycleStart.cycleRepetition</code> |
| [constr_6877] | Existence of <code>TDEventFrClusterCycleStart.frCluster</code> |
| [constr_6878] | Existence of <code>TDEventTTCancycleStart.ttCanCluster</code> |
| [constr_6879] | Existence of <code>TDEventOccurrenceExpression.formula</code> |
| [constr_6880] | Existence of <code>AutosarVariableInstance.variableInstance</code> |
| [constr_6881] | Existence of <code>AutosarOperationArgumentInstance.operationArgumentInstance</code> |
| [constr_6882] | Existence of <code>TDEventSwcInternalBehavior.tdEventSwcInternalBehaviorType</code> |
| [constr_6883] | Existence of <code>TDEventSwcInternalBehaviorRunnable</code> |
| [constr_6884] | Existence of <code>TDEventSwcInternalBehaviorReference.referencedTDEventSwc</code> |



△

| Number | Heading |
|---------------|---|
| [constr_6885] | Existence of <code>TDEventVfbPort.isExternal</code> |
| [constr_6886] | Existence of <code>TDEventVfbReference.referencedTDEventVfb</code> |
| [constr_6887] | Existence of <code>TDEventVariableDataPrototype.tdEventVariableDataPrototypeType</code> |
| [constr_6888] | Existence of <code>TDEventVariableDataPrototype.dataElement</code> |
| [constr_6889] | Existence of <code>TDEventOperation.tdEventOperationType</code> |
| [constr_6890] | Existence of <code>TDEventOperation.operation</code> |
| [constr_6891] | Existence of <code>TDEventModeDeclaration.tdEventModeDeclarationType</code> |
| [constr_6892] | Existence of <code>TDEventModeDeclaration.modeDeclaration</code> |
| [constr_6893] | Existence of <code>TDEventTrigger.tdEventTriggerType</code> |
| [constr_6894] | Existence of <code>TDEventTrigger.trigger</code> |
| [constr_6895] | Existence of <code>TimingDescriptionEventChain.response</code> |
| [constr_6896] | Existence of <code>TimingDescriptionEventChain.stimulus</code> |
| [constr_6897] | Existence of <code>TimingDescriptionEventChain.segment</code> |
| [constr_6898] | Existence of <code>ConcretePatternEventTriggering.offset</code> |
| [constr_6899] | Existence of <code>ModeInSwcInstanceRef.base</code> |

Table G.13: Added Constraints in R22-11

G.7.5 Changed Constraints in R22-11

| Number | Heading |
|---------------|--|
| [constr_4559] | Restriction of <code>TimingDescriptionEvent.category</code> |
| [constr_4560] | Restriction of <code>TimingDescriptionEventChain.category</code> |

Table G.14: Changed Constraints in R22-11

G.7.6 Deleted Constraints in R22-11

none

G.8 Change History of this document according to AUTOSAR Release R23-11

G.8.1 Added Specification Items in R23-11

| Number | Heading |
|-------------------|---|
| [TPS_TIMEX_00127] | SystemTiming of classic platform software clusters |
| [TPS_TIMEX_00128] | Default letDataExchangeParadigm |
| [TPS_TIMEX_00129] | Representation of the root in a Hierarchical Execution Order Constraint |

Table G.15: Added Specification Items in R23-11

G.8.2 Changed Specification Items in R23-11

| Number | Heading |
|-------------------|---|
| [TPS_TIMEX_00005] | Semantics of an AgeConstraint |
| [TPS_TIMEX_00009] | Optionality of Timing Extensions |
| [TPS_TIMEX_00066] | Purpose of TDCpSoftwareClusterMappingSet |
| [TPS_TIMEX_00067] | Purpose of TDCpSoftwareClusterMapping |
| [TPS_TIMEX_00068] | Purpose of TDCpSoftwareClusterResourceMapping |
| [TPS_TIMEX_00115] | Representation of an SL-LET interval <code>release</code> |
| [TPS_TIMEX_00116] | Representation of an SL-LET interval <code>terminate</code> |

Table G.16: Changed Specification Items in R23-11

G.8.3 Deleted Specification Items in R23-11

| Number | Heading |
|-------------------|--|
| [TPS_TIMEX_00037] | TimingConstraint is a Traceable |
| [TPS_TIMEX_00040] | Blueprinting VfbTiming |
| [TPS_TIMEX_00126] | Representation of an SL-LET interval in a TimingDescriptionEvent |

Table G.17: Deleted Specification Items in R23-11

G.8.4 Added Constraints in R23-11

| Number | Heading |
|---------------|---|
| [constr_6900] | Dual existence of <code>TDEventVfb.port</code> and <code>TDEventVfb.portPrototypeBlueprint</code> |
| [constr_6901] | Existence of <code>TDEventBsw.bswModuleDescription</code> |
| [constr_6906] | Conformity of <code>stimulus</code> and <code>response</code> in a <code>TimingDescriptionEventChain</code> |
| [constr_6907] | Restriction of <code>EOCExecutableEntityRefGroup.triggeringEvent</code> |
| [constr_6908] | Restriction of <code>EOCExecutableEntityRefGroup.letDataExchangeParadigm</code> |
| [constr_6909] | Singleton <code>ROOT_GROUP</code> in a Hierarchical Execution Order Constraint |
| [constr_6910] | Referencing from a <code>ROOT_GROUP</code> in a Hierarchical Execution Order Constraint |
| [constr_6911] | Referencing to a <code>ROOT_GROUP</code> in a Hierarchical Execution Order Constraint |
| [constr_6912] | Mandatory specification of LET interval recurrence |
| [constr_6913] | Restriction on <code>RTEEvent</code> s used in an LET interval |
| [constr_6914] | Restriction of the <code>port</code> context of an <code>AgeConstraint</code> |
| [constr_6915] | Affinity of <code>ISignal</code> in <code>TDEventISignal</code> |
| [constr_6916] | Affinity of <code>Frame</code> in <code>TDEventFrame</code> |
| [constr_6917] | Affinity of <code>IPdu</code> in <code>TDEventIPdu</code> |

Table G.18: Added Constraints in R23-11

G.8.5 Changed Constraints in R23-11

| Number | Heading |
|---------------|--|
| [constr_4504] | Restriction of the <code>scope</code> of an <code>AgeConstraint</code> |
| [constr_4508] | Existence of <code>TDEventVfbPort.portPrototypeBlueprint</code> |
| [constr_4523] | Restriction of <code>maxCycleRepetitions</code> and <code>maxSlotsPerCycle</code> to Repetitive Execution Order Constraint |
| [constr_4547] | Restriction of <code>ExecutionOrderConstraint.permitMultipleReferencesToEE</code> |
| [constr_4554] | Restriction of the referenced <code>TimingDescriptionEventChain</code> for a <code>letInterval</code> |
| [constr_4561] | Usage of the category value <code>DISPATCH_ENTRY_POINT</code> in <code>TimingDescriptionEvent</code> |
| [constr_4562] | Usage of the category value <code>DISPATCH_EXIT_POINT</code> in <code>TimingDescriptionEvent</code> |
| [constr_4563] | <code>TDCpSoftwareClusterMapping</code> shall reference only dispatchers or LET intervals |
| [constr_4564] | <code>TDCpSoftwareClusterResourceMapping</code> shall reference only dispatchers or LET intervals |
| [constr_4565] | Consistency of <code>TDCpSoftwareClusterMapping</code> and <code>TDCpSoftwareClusterResourceMapping</code> |



△

| Number | Heading |
|---------------|---|
| [constr_4567] | Reference provider of TDCpSoftwareClusterMapping |
| [constr_4568] | Reference requestor of TDCpSoftwareClusterMapping |

Table G.19: Changed Constraints in R23-11

G.8.6 Deleted Constraints in R23-11

| Number | Heading |
|---------------|--|
| [constr_4501] | Application rule for the occurrence expression in TDEventComplex |
| [constr_4509] | Only VfbTiming shall be a Blueprint |
| [constr_4524] | Referencing TimingDescriptionEvent |
| [constr_4533] | Usage of optional attribute ignoreOrderAllowed |
| [constr_4555] | Usage of the category value LET_RELEASE in TimingDescriptionEvent |
| [constr_4556] | Usage of the category value LET_TERMINATE in TimingDescriptionEvent |
| [constr_4557] | Usage of the category value LET_INTERVAL in TimingDescriptionEventChain |
| [constr_4558] | Applicability of LET semantics |
| [constr_4566] | SystemTiming describing timing of software clusters and category of System |
| [constr_6840] | Existence of LatencyTimingConstraint.nominal |

Table G.20: Deleted Constraints in R23-11

G.9 Change History of this document according to AUTOSAR Release R24-11

G.9.1 Added Specification Items in R24-11

none

G.9.2 Changed Specification Items in R24-11

| Number | Heading |
|-------------------|--|
| [TPS_TIMEX_00056] | Standardized categorys of TimingDescriptionEvent in Classic Platform |
| [TPS_TIMEX_00115] | Representation of an SL-LET interval release |

▽

△

| Number | Heading |
|-------------------|---|
| [TPS_TIMEX_00116] | Representation of an SL-LET interval <code>terminate</code> |

Table G.21: Changed Specification Items in R24-11

G.9.3 Deleted Specification Items in R24-11

| Number | Heading |
|-------------------|---|
| [TPS_TIMEX_00005] | Semantics of an <code>AgeConstraint</code> |
| [TPS_TIMEX_00038] | Purpose of <code>EOCExecutableEntityRefAbstract</code> |
| [TPS_TIMEX_00041] | Purpose of <code>EOCExecutableEntityRefGroup</code> |
| [TPS_TIMEX_00046] | Purpose of <code>EOCExecutableEntityRef</code> |
| [TPS_TIMEX_00047] | Purpose of <code>ExecutionOrderConstraintTypeEnum</code> |
| [TPS_TIMEX_00048] | Purpose of <code>EOCEventRef</code> |
| [TPS_TIMEX_00055] | Purpose of the attribute <code>letInterval</code> |
| [TPS_TIMEX_00066] | Purpose of <code>TDCpSoftwareClusterMappingSet</code> |
| [TPS_TIMEX_00067] | Purpose of <code>TDCpSoftwareClusterMapping</code> |
| [TPS_TIMEX_00068] | Purpose of <code>TDCpSoftwareClusterResourceMapping</code> |
| [TPS_TIMEX_00101] | Semantics of <code>accuracyInt</code> |
| [TPS_TIMEX_00102] | Optionality of <code>LatencyTimingConstraint.minimum</code> used in an LET interval |
| [TPS_TIMEX_00104] | Semantics of <code>accuracyExt</code> |
| [TPS_TIMEX_00105] | Purpose of <code>TimingClockSyncAccuracy</code> |
| [TPS_TIMEX_00106] | Purpose of <code>TimingClockSyncAccuracy.upper</code> |
| [TPS_TIMEX_00107] | Purpose of <code>TimingClockSyncAccuracy.lower</code> |
| [TPS_TIMEX_00109] | Optionality of <code>accuracy</code> |
| [TPS_TIMEX_00118] | Usage of <code>TimingClock.platformTimeBase</code> |
| [TPS_TIMEX_00119] | Purpose of <code>TDLETZoneClock</code> |
| [TPS_TIMEX_00120] | Purpose of <code>TDEventSLLET</code> |
| [TPS_TIMEX_00123] | Purpose of <code>TimingClock</code> |
| [TPS_TIMEX_00124] | Purpose of <code>TDEventSLLETPort</code> |

Table G.22: Deleted Specification Items in R24-11

G.9.4 Added Constraints in R24-11

| Number | Heading |
|---------------|---|
| [constr_6918] | Referenced <code>TimingDescriptions</code> in <code>TDCpSoftwareClusterMapping</code> and <code>TDCpSoftwareClusterResourceMapping</code> |
| [constr_6919] | Referenced <code>CpSoftwareCluster</code> of <code>TDCpSoftwareClusterMapping</code> |
| [constr_6920] | Existence of <code>LatencyTimingConstraint.minimum</code> used in an LET interval |
| [constr_6921] | Disallow <code>TimingDescriptionEventChain</code> segmental circular-referencing |

Table G.23: Added Constraints in R24-11

G.9.5 Changed Constraints in R24-11

| Number | Heading |
|---------------|--|
| [constr_4500] | Restricted usage of Occurrence Expression functions |
| [constr_4502] | Use references only as function operands |
| [constr_4503] | Restricted usage of <code>AutosarOperationArgumentInstance</code> for Content Filter |
| [constr_4504] | Restriction of the <code>scope</code> of an <code>AgeConstraint</code> |
| [constr_4505] | Specifying minimum and maximum number of occurrences |
| [constr_4506] | Specifying minimum inter-arrival time and pattern length |
| [constr_4507] | Specifying pattern length, pattern jitter and pattern period |
| [constr_4508] | Existence of <code>TDEventVfbPort.portPrototypeBlueprint</code> |
| [constr_4510] | Specifying references to <code>RunnableEntity</code> and <code>VariableAccess</code> |
| [constr_4511] | Validity of referencing <code>RunnableEntity</code> |
| [constr_4512] | Validity of referencing <code>VariableAccess</code> |
| [constr_4513] | <code>SynchronizationTimingConstraint</code> shall reference at least two events |
| [constr_4514] | <code>SynchronizationTimingConstraint</code> shall reference at least two event chains |
| [constr_4515] | Orthogonality of <code>stimulus</code> and <code>response</code> in a <code>TimingDescriptionEventChain</code> |
| [constr_4516] | Completeness of a composed <code>TimingDescriptionEventChain</code> |
| [constr_4518] | Specifying end-points of a composed <code>TimingDescriptionEventChain</code> |
| [constr_4519] | Specifying <code>patternLength</code> |
| [constr_4520] | Specifying attribute <code>synchronizationConstraintType</code> |
| [constr_4521] | Specifying attribute <code>synchronizationConstraintType</code> |
| [constr_4522] | <code>SynchronizationTimingConstraint</code> shall either reference events or event chains |
| [constr_4523] | Restriction of <code>maxCycleRepetitions</code> and <code>maxSlotsPerCycle</code> to Repetitive Execution Order Constraint |
| [constr_4525] | Precedence of successor relationships <code>successor</code> and <code>directSuccessor</code> |
| [constr_4526] | Specifying <code>maxCycles</code> and <code>maxSlots</code> in a Repetitive Execution Order Constraint |





| Number | Heading |
|---------------|--|
| [constr_4527] | Referencing <code>TimingDescriptionEvent</code> in a Repetitive Execution Order Constraint |
| [constr_4528] | The <i>root</i> <code>EOCExecutableEntityRefGroup</code> shall reference only <code>EOCExecutableEntityRefGroup</code> s |
| [constr_4529] | Number of nested elements referenced by the <i>root</i> <code>EOCExecutableEntityRefGroup</code> |
| [constr_4530] | An <code>EOCExecutableEntityRefGroup</code> representing a cycle shall reference only <code>EOCExecutableEntityRef</code> s respectively <code>EOCEventRefs</code> |
| [constr_4531] | Number of nested elements referenced by <code>EOCExecutableEntityRefGroup</code> representing a cycle |
| [constr_4532] | Successor relationship is not self-referencing |
| [constr_4533] | Maximum number of successor relationships |
| [constr_4534] | Maximum number of <code>directSuccessor</code> relationships |
| [constr_4536] | Compatible recurrence of any <code>ExecutableEntity</code> |
| [constr_4537] | References among elements in an <code>ExecutionOrderConstraint</code> |
| [constr_4538] | Hierarchical Execution Order Constraint: <code>EOCExecutableEntityRef</code> , <code>EOCEventRef</code> , and <code>EOCExecutableEntityRefGroup</code> shall be target or source of a successor relationship |
| [constr_4539] | The successor relationships <code>successor</code> and <code>directSuccessor</code> shall not be used |
| [constr_4540] | <code>maxCycles</code> and <code>maxSlots</code> shall not be zero |
| [constr_4541] | Existence of <code>EOCExecutableEntityRef.executable</code> in an Ordinary Execution Order Constraint |
| [constr_4542] | Existence of <code>EOCExecutableEntityRef.executable</code> in a Hierarchical Execution Order Constraint |
| [constr_4543] | Maximum value of <code>minimumInterArrivalTime</code> |
| [constr_4544] | Specifying <code>patternLength</code> , <code>patternJitter</code> and <code>patternPeriod</code> |
| [constr_4545] | Referring either <code>ExecutableEntity</code> s or <code>AbstractEvents</code> |
| [constr_4546] | Setting the attribute <code>isEvent</code> |
| [constr_4547] | Restriction of <code>ExecutionOrderConstraint.permitMultipleReferencesToEE</code> |
| [constr_4548] | Existence of <code>EOCEventRef.event</code> in an Ordinary Execution Order Constraint |
| [constr_4549] | Existence of <code>EOCEventRef.event</code> in a Hierarchical Execution Order Constraint |
| [constr_4551] | Use only Numericals in <code>TDEventOccurrenceExpression</code> |
| [constr_4552] | Restricted usage of <code>AutosarVariableInstance</code> for Content Filter |
| [constr_4554] | Restriction of the referenced <code>TimingDescriptionEventChain</code> for a <code>letInterval</code> |
| [constr_4559] | Restriction of <code>TimingDescriptionEvent.category</code> |
| [constr_4561] | Usage of the category value <code>DISPATCH_ENTRY_POINT</code> in <code>TimingDescriptionEvent</code> |
| [constr_4562] | Usage of the category value <code>DISPATCH_EXIT_POINT</code> in <code>TimingDescriptionEvent</code> |
| [constr_4565] | Consistency of <code>TDCpSoftwareClusterMapping.timingDescription</code> and <code>TDCpSoftwareClusterResourceMapping.timingDescription</code> |





| Number | Heading |
|---------------|--|
| [constr_6816] | Restricted usage of <code>TimingDescriptionEventChain.isPipeliningPermitted</code> in <code>TimingDescriptionEventChain</code> |
| [constr_6817] | Restricted usage of <code>TimingDescriptionEvent.clockReference</code> |
| [constr_6818] | Existence of <code>EventTriggeringConstraint.event</code> |
| [constr_6819] | Existence of <code>PeriodicEventTriggering.jitter</code> |
| [constr_6820] | Existence of <code>PeriodicEventTriggering.minimumInterArrivalTime</code> |
| [constr_6821] | Existence of <code>PeriodicEventTriggering.period</code> |
| [constr_6822] | Existence of <code>SporadicEventTriggering.maximumInterArrivalTime</code> |
| [constr_6823] | Existence of <code>SporadicEventTriggering.minimumInterArrivalTime</code> |
| [constr_6824] | Existence of <code>ConcretePatternEventTriggering.patternLength</code> |
| [constr_6825] | Existence of <code>BurstPatternEventTriggering.maxNumberOfOccurrences</code> |
| [constr_6826] | Existence of <code>BurstPatternEventTriggering.minimumInterArrivalTime</code> |
| [constr_6827] | Existence of <code>BurstPatternEventTriggering.patternLength</code> |
| [constr_6828] | Existence of <code>ArbitraryEventTriggering.minimumDistance</code> |
| [constr_6829] | Existence of <code>ArbitraryEventTriggering.maximumDistance</code> |
| [constr_6830] | Existence of <code>ConfidenceInterval.lowerBound</code> |
| [constr_6831] | Existence of <code>ConfidenceInterval.probability</code> |
| [constr_6832] | Existence of <code>ConfidenceInterval.upperBound</code> |
| [constr_6833] | Existence of <code>ExecutionOrderConstraint.orderedElement</code> |
| [constr_6834] | Existence of <code>EOCExecutableEntityRefGroup.nestedElement</code> |
| [constr_6835] | Existence of <code>ExecutionTimeConstraint.executionTimeType</code> |
| [constr_6836] | Existence of <code>ExecutionTimeConstraint.executable</code> |
| [constr_6837] | Existence of <code>LatencyTimingConstraint.latencyConstraintType</code> |
| [constr_6838] | Existence of <code>LatencyTimingConstraint.maximum</code> |
| [constr_6839] | Existence of <code>LatencyTimingConstraint.minimum</code> |
| [constr_6841] | Existence of <code>LatencyTimingConstraint.scope</code> |
| [constr_6842] | Existence of <code>OffsetTimingConstraint.maximum</code> |
| [constr_6843] | Existence of <code>OffsetTimingConstraint.minimum</code> |
| [constr_6844] | Existence of <code>OffsetTimingConstraint.source</code> |
| [constr_6845] | Existence of <code>OffsetTimingConstraint.target</code> |
| [constr_6846] | Existence of <code>SynchronizationTimingConstraint.synchronizationConstraintType</code> |
| [constr_6847] | Existence of <code>SynchronizationTimingConstraint.tolerance</code> |
| [constr_6848] | Existence of <code>VfbTiming.component</code> |
| [constr_6849] | Existence of <code>SystemTiming.system</code> |
| [constr_6850] | Existence of <code>BswModuleTiming.behavior</code> |
| [constr_6851] | Existence of <code>BswCompositionTiming.implementation</code> |
| [constr_6852] | Existence of <code>EcuTiming.ecuConfiguration</code> |
| [constr_6853] | Existence of <code>ModeInBswInstanceRef.contextModeDeclarationGroupPrototype</code> |



△

| Number | Heading |
|---------------|---|
| [constr_6854] | Existence of <code>ModeInBswInstanceRef.targetModeDeclaration</code> |
| [constr_6855] | Existence of <code>ModeInSwcInstanceRef.contextModeDeclarationGroupPrototype</code> |
| [constr_6856] | Existence of <code>ModeInSwcInstanceRef.contextPort</code> |
| [constr_6857] | Existence of <code>ModeInSwcInstanceRef.targetModeDeclaration</code> |
| [constr_6858] | Existence of <code>TDEventBswInternalBehavior.tdEventBswInternalBehaviorType</code> |
| [constr_6859] | Existence of <code>TDEventBswInternalBehavior.bswModuleEntity</code> |
| [constr_6860] | Existence of <code>TDEventBswModule.tdEventBswModuleType</code> |
| [constr_6861] | Existence of <code>TDEventBswModule.bswModuleEntry</code> |
| [constr_6862] | Existence of <code>TDEventBswModeDeclaration.tdEventBswModeDeclarationType</code> |
| [constr_6863] | Existence of <code>TDEventBswModeDeclaration.modeDeclaration</code> |
| [constr_6864] | Existence of <code>TDEventISignal.tdEventType</code> |
| [constr_6865] | Existence of <code>TDEventISignal.iSignal</code> |
| [constr_6866] | Existence of <code>TDEventISignal.physicalChannel</code> |
| [constr_6867] | Existence of <code>TDEventIPdu.tdEventType</code> |
| [constr_6868] | Existence of <code>TDEventIPdu.iPdu</code> |
| [constr_6869] | Existence of <code>TDEventIPdu.physicalChannel</code> |
| [constr_6870] | Existence of <code>TDEventFrame.tdEventType</code> |
| [constr_6871] | Existence of <code>TDEventFrame.frame</code> |
| [constr_6872] | Existence of <code>TDEventFrame.physicalChannel</code> |
| [constr_6873] | Existence of <code>TDEventFrameEthernet.tdEventType</code> |
| [constr_6874] | Existence of <code>TDHeaderIdRange.maxHeaderId</code> |
| [constr_6875] | Existence of <code>TDHeaderIdRange.minHeaderId</code> |
| [constr_6876] | Existence of <code>TDEventCycleStart.cycleRepetition</code> |
| [constr_6877] | Existence of <code>TDEventFrClusterCycleStart.frCluster</code> |
| [constr_6878] | Existence of <code>TDEventTTCANCycleStart.ttCanCluster</code> |
| [constr_6879] | Existence of <code>TDEventOccurrenceExpression.formula</code> |
| [constr_6880] | Existence of <code>AutosarVariableInstance.variableInstance</code> |
| [constr_6881] | Existence of <code>AutosarOperationArgumentInstance.operationArgumentInstance</code> |
| [constr_6882] | Existence of <code>TDEventSwcInternalBehavior.tdEventSwcInternalBehaviorType</code> |
| [constr_6883] | Existence of <code>TDEventSwcInternalBehavior.runnable</code> |
| [constr_6884] | Existence of <code>TDEventSwcInternalBehaviorReference.referencedTDEventSwc</code> |
| [constr_6885] | Existence of <code>TDEventVfbPort.isExternal</code> |
| [constr_6886] | Existence of <code>TDEventVfbReference.referencedTDEventVfb</code> |
| [constr_6887] | Existence of <code>TDEventVariableDataPrototype.tdEventVariableDataPrototypeType</code> |

▽

△

| Number | Heading |
|---------------|---|
| [constr_6888] | Existence of <code>TDEventVariableDataPrototype.dataElement</code> |
| [constr_6889] | Existence of <code>TDEventOperation.tdEventOperationType</code> |
| [constr_6890] | Existence of <code>TDEventOperation.operation</code> |
| [constr_6891] | Existence of <code>TDEventModeDeclaration.tdEventModeDeclarationType</code> |
| [constr_6892] | Existence of <code>TDEventModeDeclaration.modeDeclaration</code> |
| [constr_6893] | Existence of <code>TDEventTrigger.tdEventTriggerType</code> |
| [constr_6894] | Existence of <code>TDEventTrigger.trigger</code> |
| [constr_6895] | Existence of <code>TimingDescriptionEventChain.response</code> |
| [constr_6896] | Existence of <code>TimingDescriptionEventChain.stimulus</code> |
| [constr_6897] | Existence of <code>TimingDescriptionEventChain.segment</code> |
| [constr_6898] | Existence of <code>ConcretePatternEventTriggering.offset</code> |
| [constr_6899] | Existence of <code>ModeInSwcInstanceRef.base</code> |
| [constr_6900] | Dual existence of <code>TDEventVfb.port</code> and <code>TDEventVfb.portPrototypeBlueprint</code> |
| [constr_6901] | Existence of <code>TDEventBsw.bswModuleDescription</code> |
| [constr_6906] | Conformity of <code>stimulus</code> and <code>response</code> in a <code>TimingDescriptionEventChain</code> |
| [constr_6907] | Restriction of <code>EOCExecutableEntityRefGroup.triggeringEvent</code> |
| [constr_6908] | Restriction of <code>EOCExecutableEntityRefGroup.letDataExchangeParadigm</code> |
| [constr_6909] | Singleton <code>ROOT_GROUP</code> in a Hierarchical Execution Order Constraint |
| [constr_6910] | Referencing from a <code>ROOT_GROUP</code> in a Hierarchical Execution Order Constraint |
| [constr_6911] | Referencing to a <code>ROOT_GROUP</code> in a Hierarchical Execution Order Constraint |
| [constr_6912] | Mandatory specification of LET interval recurrence |
| [constr_6913] | Restriction on <code>RTEEvents</code> used in an LET interval |
| [constr_6914] | Restriction of the <code>port</code> context of an <code>AgeConstraint</code> |
| [constr_6915] | Affinity of <code>ISignal</code> in <code>TDEventISignal</code> |
| [constr_6916] | Affinity of <code>Frame</code> in <code>TDEventFrame</code> |
| [constr_6917] | Affinity of <code>IPdu</code> in <code>TDEventIPdu</code> |

Table G.24: Changed Constraints in R24-11

▽

G.9.6 Deleted Constraints in R24-11

| Number | Heading |
|---------------|---|
| [constr_4517] | Referencing no further event chain <code>segments</code> |
| [constr_4550] | A Hierarchical Execution Order Constraint shall have an unambiguous root <code>EOCExecutableEntityRefGroup</code> |
| [constr_4560] | Restriction of <code>TimingDescriptionEventChain.category</code> |

△

| Number | Heading |
|---------------|--|
| [constr_4563] | TDCpSoftwareClusterMapping shall reference only dispatchers or LET intervals |
| [constr_4564] | TDCpSoftwareClusterResourceMapping shall reference only dispatchers or LET intervals |
| [constr_4567] | Reference provider of TDCpSoftwareClusterMapping |
| [constr_4568] | Reference requestor of TDCpSoftwareClusterMapping |

Table G.25: Deleted Constraints in R24-11