Document Title	Collection of constraints on AUTOSAR M1 models
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	635

Document Status	published
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	R22-11

Document Change History			
Date	**Release**	**Changed by**	**Description**
2022-11-24	R22-11	AUTOSAR Release Management	• Updated constraints according to changes in TPS documents
2021-11-25	R21-11	AUTOSAR Release Management	• Updated constraints according to changes in TPS documents
2020-11-30	R20-11	AUTOSAR Release Management	• Updated constraints according to changes in TPS documents • Removed all SWS constraints • Split document into 3 documents, reflecting the standards CP, AP, FO
2019-11-28	R19-11	AUTOSAR Release Management	• Updated constraints according to changes in SWS and TPS documents • Changed Document Status from Final to published
2018-10-31	4.4.0	AUTOSAR Release Management	Completion of constraint context by adding tables and classtables referenced by model constraints to this document
2017-12-08	4.3.1	AUTOSAR Release Management	minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
2016-11-30	4.3.0	AUTOSAR Release Management	minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation

2015-07-31	4.2.2	AUTOSAR Release Management	minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
2014-10-31	4.2.1	AUTOSAR Release Management	Editorial changes
2013-10-31	4.1.2	AUTOSAR Release Management	Updated constraints according to changes in SWS and TPS documents
2013-03-15	4.1.1	AUTOSAR Administration	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

References

[1] List of Basic Software Modules
AUTOSAR_TR_BSWModuleList

[2] Software Component Template
AUTOSAR_TPS_SoftwareComponentTemplate

[3] Specification of RTE Software
AUTOSAR_SWS_RTE

[4] Unified diagnostic services (UDS) – Part 1: Specification and requirements (Release 2006-12)
http://www.iso.org

[5] Road vehicles – End-of-life activation of on-board pyrotechnic devices – Part 2: Communication requirements
http://www.iso.org

[6] ISO 17356-4: Road vehicles – Open interface for embedded automotive applications – Part 4: OSEK/VDX Communication (COM)

[7] Collection of blueprints for AUTOSAR M1 models
AUTOSAR_MOD_GeneralBlueprints

[8] Information technology – Universal Coded Character Set (UCS)
http://www.iso.org

[9] Generic Structure Template
AUTOSAR_TPS_GenericStructureTemplate

[10] XML Path language (XPath)
http://www.w3.org/TR/xpath/

[11] Specification of COM Based Transformer
AUTOSAR_SWS_COMBasedTransformer

[12] SAE J1939-21 Data Link Layer

[13] Transport Layer Security (TLS) Parameters
https://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml

Document ID 635: AUTOSAR_TR_ClassicAutosarModelConstraints

1 Document Information and Content

This auxiliary document provides a collection of constraints for AUTOSAR models. All constraints are copied from template specification from the AUTOSAR Classic Platform, so this document does not introduce any new constraints.

A list of the documents that the constraints originate from can be found in the table of contents. Chapter 2 contains the collected constraints, grouped by source documents. All constraints from the same source document are contained within a single section.

2 Autosar Model Constraints

2.1 TPS_BSWModuleDescriptionTemplate

[constr_1275] **Applicability of reference `startsOnEvent` for `BswScheduleEvent`** ⌈The reference `BswScheduleEvent.startsOnEvent` shall only refer to a `BswSchedulableEntity`.

⌋*()*

[constr_1276] Applicability of reference `startsOnEvent` for `BswOperationInvokedEvent` ⌈The reference `BswOperationInvokedEvent.startsOnEvent` shall only refer to a `BswCalledEntity`.

⌋*()*

[constr_4013] BSW service identifier ⌈For Standardized Interfaces, this identifier is defined in the AUTOSAR Software Specification (SWS) of the module. In case the C-function prototype represented by the entry is not standardized, it still can be used optionally, but its value shall differ from the standardized ones.

⌋*()*

[constr_4014] Call type and execution context ⌈Within a given `BswModuleEntry`, the following constraint holds for its attributes:

- `callType`=='`interrupt`' is not allowed together with `executionContext`=='`task`' or =='`hook`'

- `callType`=='`scheduled`' is not allowed together with `executionContext`=='`interruptCat1`' or =='`interruptCat2`'

- other combinations of these two enums are allowed

⌋*()*

[constr_4015] **`calledEntry` constraints for direct calls** ⌈The following holds if `callPoint` is aggregated as an instance of `BswDirectCallPoint`:

- `BswModuleEntity.callPoint.calledEntry.executionContext` shall be identical to `BswModuleEntity.implementedEntry.executionContext`

- `BswModuleEntity.callPoint.calledEntry.callType` shall have the value '`regular`' or '`callback`'

⌋*()*

[constr_4016] `BswCalledEntity` constraints ⌈

- `BswCalledEntity.implementedEntry.callType` shall be '`regular`' or '`callback`'

- `BswCalledEntity`.`implementedEntry`.`executionContext` is in general not restricted, but see [constr_4076] for constraints on the server side of a Client-Server communication.

⌋*()*

[constr_4017] `BswSchedulableEntity` constraints ⌈

- `BswModuleEntity`.`implementedEntry`.`callType` shall be `'scheduled'`

- `BswModuleEntity`.`implementedEntry`.`executionContext` shall be `'task'`

⌋*()*

[constr_4018] `BswInterruptEntity` constraints ⌈

- `BswInterruptEntity`.`implementedEntry`.`callType` shall be `'interrupt'`

- `BswInterruptEntity`.`implementedEntry`.`executionContext` shall be `'interruptCat1'` if and only if `BswInterruptEntity`.`interruptCategory` is `'Cat1'`

- `BswInterruptEntity`.`implementedEntry`.`executionContext` shall be `'interruptCat2'` if and only if `BswInterruptEntity`.`interruptCategory` is `'Cat2'`

⌋*()*

[constr_4019] BSW module identifier ⌈`BswModuleDescription`.`moduleId` shall refer to the identifier of the standardized AUTOSAR modules according to [1], if applicable[1]. Otherwise (e.g. for ICC2 clusters) the identifier shall either be empty or chosen differently from the ones given in [1].

⌋*()*

[constr_4020] Allowed categories of `BswModuleDescription` ⌈

category	Explanation
BSW_MODULE	Specifies a single BSW module (ICC3 granularity).
BSW_CLUSTER	Specifies a BSW module cluster (ICC2 granularity).
LIBRARY	Specifies a Library (not restricted to be used within the BSW).

⌋*()*

[constr_4021] Implementation policy of function pointer target ⌈
A `BswModuleEntry` can only be used as target of a function pointer (`SwPoint-`

---

[1]Note that there may be more than one module in an ECU software with the same identifier, e.g. according to the standard Complex Drivers all have the same identifier.

`erTargetProps.functionPointerSignature`), if its `swServiceImplPolicy` is
'`standard`'.

⌋*()*

[constr_4022] `BswModuleEntity` only uses the module's interface ⌈

- `BswModuleEntity.implementedEntry` shall refer to an element declared as
  `implementedEntry` of the enclosing `BswModuleDescription`

- `BswModuleEntity.callPoint.calledEntry` - where `callPoint` is instan-
  tiated from `BswDirectCallPoint` - shall refer to an element declared as `ex-`
  `pectedEntry` or `implementedEntry` of the enclosing `BswModuleDescrip-`
  `tion`.

- `BswModuleEntity.callPoint.calledEntry` - where `callPoint` is
  instantiated from `BswSynchronousServerCallPoint` or `BswAsyn-`
  `chronousServerCallPoint` - shall refer to an element declared as
  `requiredClientServerEntry` of the enclosing `BswModuleDescription`.

- `BswModuleEntity.callPoint` - where `callPoint` is instantiated from
  `BswAsynchronousServerCallResultPoint` - shall refer to an `BswAsyn-`
  `chronousServerCallPoint` declared in turn as `callPoint` of the same
  `BswModuleEntity`.

- `BswModuleEntity.issuedTrigger` shall refer to an element declared as `re-`
  `leasedTrigger` of the enclosing `BswModuleDescription`

- `BswModuleEntity.managedModeGroup` shall refer to an element declared as
  `providedModeGroup` of the enclosing `BswModuleDescription`

- `BswModuleEntity.accessedModeGroup` shall refer to an element declared
  as `requiredModeGroup` of the enclosing `BswModuleDescription`

- `BswModuleEntity.dataSendPoint.accessedVariable` shall refer to an el-
  ement declared as `providedData` of the enclosing `BswModuleDescription`

- `BswModuleEntity.dataReceivePoint.accessedVariable` shall refer to
  an element declared as `requiredData` of the enclosing `BswModuleDescrip-`
  `tion`

- an `accessedModeGroup` should be allowed to refer to an element declared as
  `providedModeGroup`

⌋*()*

[constr_4023] External trigger shall belong to the interface ⌈A `BswExternal-`
`TriggerOccurredEvent` shall refer to a `Trigger` that is declared via `BswMod-`
`uleDescription.requiredTrigger` for the same module.

⌋*()*

[constr_4024] Semantics of BSW mode switch event ⌈If `BswModeSwitchEvent.activation` has the value `onTransition BswModeSwitchEvent` shall refer to two different modes belonging to the same instance of ModeDeclarationGroup, their order defining the direction of the transition. In all other cases, `BswModeSwitchEvent` shall refer to exactly one mode.

⌋*()*

[constr_4025] Modes used by BSW mode switch event ⌈The `ModeDeclaration` used by `BswModeSwitchEvent` shall belong to the `ModeDeclarationGroupPrototype` referred as `BswInternalBehavior.entity.accessedModeGroup` of the enclosing `BswInternalBehavior`.

⌋*()*

[constr_4026] Mode group used by BSW mode switch acknowledge event ⌈The `ModeDeclarationGroupPrototype` used by `BswModeSwitchedAckEvent` shall be referred as `BswModuleDescription.providedModeGroup` by the same module.

⌋*()*

[constr_4028] Semantics of memory section type ⌈`sectionType` shall be semantically compatible to the usage of the enclosing `SwAddrMethod`, this means especially that if `SwAddrMethod` is associated by `ExecutableEntity`-s, the `sectionType` shall be usable as code section, if it is associated by `SwDataDefProps`, `sectionType` shall be usable as data section.

⌋*()*

[constr_4029] Measured stack usage ⌈The attribute values of `MeasuredStackUsage` shall fulfill:

`minimumMemoryConsumption` <= `averageMemoryConsumption` <= `maximumMemoryConsumption`

⌋*()*

[constr_4030] Measured heap usage ⌈The attribute values of `MeasuredHeapUsage` shall fulfill:

`minimumMemoryConsumption` <= `averageMemoryConsumption` <= `maximumMemoryConsumption`

⌋*()*

[constr_4031] Analyzed execution time ⌈The attribute values of `AnalyzedExecutionTime` shall fulfill:

`bestCaseExecutionTime` <= `bestCaseExecutionTime`

⌋*()*

[constr_4032] Measured execution time ⌈The attribute values of `MeasuredExecutionTime` shall fulfill:

`minimumExecutionTime` <= `nominalExecutionTime` <= `maximumExecution-Time`

⌋*()*

[constr_4033] Simulated execution time ⌈The attribute values of `SimulatedExecutionTime` shall fulfill:

`minimumExecutionTime` <= `nominalExecutionTime` <= `maximumExecution-Time`

⌋*()*

[constr_4034] Target and context of MC emulation reference ⌈Within one `ImplementationElementInParameterInstanceRef`, the `target` shall refer to a sub-element of the `ParameterDataPrototype` which is referred as `context`.

⌋*()*

[constr_4038] `bswModuleDependency` shall refer to a different module ⌈

- `BswModuleDescription.bswModuleDependency.targetModuleId` (if given) shall differ from `BswModuleDescription.moduleId`. This does not hold if the value is 254 (used for IO Hardware Abstraction modules) or 255 (used for Complex Driver modules).

- `BswModuleDependency.targetModuleRef` (if given) shall differ from the package location of the `BswModuleDescription` that owns the `BswModuleDependency`.

⌋*()*

[constr_4039] Semantics of `SwcBswMapping` ⌈An `SwcBswMapping` is only valid, if the referred `SwcInternalBehavior` is aggregated by a `ServiceSwComponentType`, `EcuAbstractionSwComponentType` or `ComplexDeviceDriverSwComponentType`.

⌋*()*

[constr_4040] Synchronized mode groups shall have same type ⌈`SwcBswSynchronizedModeGroupPrototype` can only refer to equally typed `ModeDeclarationGroupPrototype`s, i.e. which have identical `ModeDeclarationGroup`s.

⌋*()*

[constr_4041] Synchronized mode groups shall have same context ⌈The mapping defined by `SwcBswSynchronizedModeGroupPrototype` implies that the component providing the one mode group prototype is also mapped to the module which provides the other mode group prototype by means of synchronizing their respective behaviors in `SwcBswMapping`.

⌋*()*

[constr_4042] Synchronized triggers shall have same context ⌈The mapping defined by `SwcBswSynchronizedTrigger` implies that the component providing the one trigger is also mapped to the module which provides the other trigger by means of synchronizing their respective behaviors in `SwcBswMapping`.

⌋*()*

[constr_4043] Period of `BswTimingEvent` ⌈`BswTimingEvent.period` shall be greater than 0.

⌋*()*

[constr_4044] Content of `McSwEmulationMethodSupport` ⌈The following constraints hold for the attributes of `McSwEmulationMethodSupport`:

- If `category` is DOUBLE_POINTERED, a `baseReference` shall exist.
- If `category` is SINGLE_POINTERED, a `referenceTable` shall exist.
- If `category` is INITIALIZED_RAM, one or more `elementGroup`s shall exist.

⌋*()*

[constr_4045] `implementationConfigVariant` of preconfigured configuration ⌈An `EcucModuleConfigurationValues` element with the `implementationConfigVariant` set to the value PreconfiguredConfiguration shall only be referenced in the role `preconfiguredConfiguration` and no other value for `implementationConfigVariant` is allowed in this role.

⌋*()*

[constr_4046] `implementationConfigVariant` of recommended configuration ⌈An `EcucModuleConfigurationValues` element with the `implementationConfigVariant` set to the value RecommendedConfiguration shall only be referenced in the role `recommendedConfiguration` and no other value for `implementationConfigVariant` is allowed in this role.

⌋*()*

[constr_4047] Multiplicity of vendor specific configuration parameters ⌈The association `BswImplementation.vendorSpecificModuleDef` shall be implemented as reference to one or more instances of `EcucModuleDef` if the underlying `BswModuleDescription` has the `category` BSW_CLUSTER. In all other cases, it shall refer to exactly one instance of `EcucModuleDef` (the one belonging to this module).

⌋*()*

[constr_4048] Multiplicity of preconfigured values ⌈The association `BswImplementation.preconfiguredConfiguration` shall be implemented as reference to zero or more different instances of `EcucModuleConfigurationValues` if the underlying `BswModuleDescription` has the `category` BSW_CLUSTER. In all other

cases, it shall refer to at most one instance of `EcucModuleConfigurationValues` (the one belonging to this module).

⌋*()*

[constr_4051] `RoleBasedDataAssignment` **in BSW** ⌈When used in the context of `BswServiceDependency`, the following restriction hold for date references described by `RoleBasedDataAssignment`:

- Within `RoleBasedDataAssignment.usedDataElement`, only the reference `AutosarVariableRef.localVariable` is applicable.

- Within `RoleBasedDataAssignment.usedParameterElement`, only the reference `AutosarParameterRef.localParameter` is applicable.

- The reference `RoleBasedDataAssignment.usedPim` shall not be set.

⌋*()*

[constr_4052] `BswModuleEntry` **returnType direction** ⌈
`BswModuleEntry.returnType.direction` shall not have the value **in** or **inout**.

⌋*()*

[constr_4053] `BswModuleEntry` **argument direction** ⌈
If `BswModuleEntry.argument.direction` has the value **out** or **inout**, the corresponding `BswModuleEntry.argument.swDataDefProps` plus eventually referred `ImplementationDataType` shall be such that they result in a pointer declaration.

⌋*()*

[constr_4054] Unambiguous links to addressing method ⌈`MemorySection.executableEntity` shall not be defined, if `MemorySection.swAddrMethod` represents a data section. `MemorySection.executableEntity` shall not refer to an `ExecutableEntity` which is linked to a different `SwAddrMethod` than `MemorySection.swAddrMethod`.

⌋*()*

[constr_4056] `BswModuleEntry` **with no** `returnType` ⌈
In case of an empty return type ("void" in C) the reference `BswModuleEntry.returnType` shall not be set.

⌋*()*

[constr_4057] `BswModuleEntry` **with no argument** ⌈
In case of an empty argument list ("void" in C) no reference `BswModuleEntry.argument` shall be set.

⌋*()*

[constr_4058] Different mode groups in mapped BSWM and SWC shall have different names ⌈If an `SwcInternalBehavior` is mapped to a `BswInternalBehavior` the corresponding SWC and BSW module descriptions may not refer to different

`ModeDeclarationGroup`s having the same `shortName` but different elements. This holds especially if these mode groups are not synchronized but used independently.

⌋*()*

[constr_4059] Different mode groups referred by a BSWM shall have different names ⌈A `BswModuleDescription` may not refer to different `ModeDeclarationGroup`s (via `requiredModeGroup` and/or `providedModeGroup`) having the same `shortName` but different elements.

⌋*()*

[constr_4060] Allowed values of `Trigger.swImplPolicy` for BSW ⌈The **only** allowed values for the attribute `Trigger.swImplPolicy` are either STANDARD (in which case the `Trigger` processing does not use a queue) or QUEUED (in which case the processing of `Trigger`s positively uses a queue).

⌋*()*

[constr_4061] Completeness of MC emulation reference ⌈If an `McDataInstance` in the role of a `subElement` of another `McDataInstance` specifies an `instanceInMemory`, then the containing `McDataInstance` shall also specify an `instanceInMemory`. The `target` of the latter (i.e. upper level) `instanceInMemory` shall be identical (including array index, if defined) to the `context` of the first (i.e. lower level) `instanceInMemory`.

⌋*()*

[constr_4062] Mandatory symbol for `McDataInstance` root ⌈`McDataInstance`s directly aggregated in `McSupportData` shall have a valid `McDataInstance.symbol`.

⌋*()*

[constr_4063] Restrictions of `ModeRequestTypeMap` in BSW ⌈For every `ModeDeclarationGroup` referenced by a `ModeDeclarationGroupPrototype` used in a `BswModuleDescription` a `ModeRequestTypeMap` shall exist that points to the `ModeDeclarationGroup` and also to an eligible `ImplementationDataType`.

The `ModeRequestTypeMap` shall be aggregated by a `DataTypeMappingSet` which is referenced from the `BswInternalBehavior` that is aggregated by the `BswModuleDescription`.

⌋*()*

[constr_4064] Synchronized triggers shall implement same policy ⌈The mapping defined by `SwcBswSynchronizedTrigger` is only valid if the attribute `SwcBswSynchronizedTrigger.swcTrigger.swImplPolicy` has the same value as the attribute `SwcBswSynchronizedTrigger.bswTrigger.swImplPolicy`.

⌋*()*

[constr_4065] Allowed values of `BswInternalTriggeringPoint.swImplPolicy` ⌈The **only** allowed values for the attribute `BswInternalTriggeringPoint.swImplPolicy` are either `STANDARD` (in which case the internal trigger processing does not use a queue) or `QUEUED` (in which case the internal trigger processing uses a queue).

⌋*()*

[constr_4066] `BswModeSwitchEvent` and the definition of `ModeTransition` ⌈For each pair of `ModeDeclaration`s referenced by a `BswModeSwitchEvent` with attribute `activation` set to `onTransition` a `ModeTransition` shall be defined in the corresponding direction (i.e. from `exitedMode` to `enteredMode`). This constraint shall only apply if the respective `ModeDeclarationGroup` defines at least one `modeTransition`.

⌋*()*

[constr_4068] `McFunctionDataRefSet.flatMapEntry`'s semantic ⌈

- An `McFunctionDataRefSet` aggregated in the role of `McFunction.defCalprmSet` or `McFunction.refCalprmSet` shall only refer to `FlatInstanceDescriptor`s that
    - either can be traced down to a `ParameterDataPrototype`
    - or can be traced down to a `VariableDataPrototype` of category `COM_AXIS`, `RES_AXIS`, `CURVE`, `MAP`, `CUBOID`, `CUBE_4`, `CUBE_5` or `VAL_BLK`

  and which are declared for calibration access i.e. have an associated `SwDataDefProps.swCalibrationAccess` set to `readWrite` or `readOnly`.

- An `McFunctionDataRefSet` aggregated in the role of `McFunction.inMeasurementSet`, `McFunction.outMeasurementSet` or `McFunction.locMeasurementSet` shall only refer to `FlatInstanceDescriptor`s that can be traced down to either a `VariableDataPrototype`, an `ArgumentDataPrototype` or a `ModeDeclarationGroupPrototype` and are declared as measurable i.e. have an associated `SwDataDefProps.swCalibrationAccess` set to `readOnly`.

⌋*()*

[constr_4069] `McFunctionDataRefSet.mcDataInstance`'s semantic ⌈

- An `McFunctionDataRefSet` aggregated in the role of `McFunction.defCalprmSet` or `McFunction.refCalprmSet` shall only refer to `McDataInstance`s that are declared for calibration access i.e. are aggregated in the role `McSupportData.mcParameterInstance` or `McSupportData.mcVariableInstance` of category `COM_AXIS`, `RES_AXIS`, `CURVE`, `MAP`, `CUBOID`, `CUBE_4`, `CUBE_5` or `VAL_BLK`.

- An `McFunctionDataRefSet` aggregated in the role of `McFunction.inMeasurementSet`, `McFunction.outMeasurementSet` or `McFunction.locMea-`

surementSet shall only refer to `McDataInstance`s that are declared as measurable i.e. are aggregated in the role `McSupportData`.`mcVariableInstance`.

⌋*()*

[constr_4070] Applicability of `BswModuleEntity.activationReason` ⌈An `activationReason` shall not be set

- for instances of `BswInterruptEntity`

- for instances of `BswCalledEntity`

⌋*()*

[constr_4071] Synchronized runnables and schedulable entities shall be consistent ⌈In the case that a `RunnableEntity` is mapped to a `BswCalledEntity` or `BswSchedulableEntity` the RTE Generator emits an Entry Point Prototype only for the `BswCalledEntity` or the `BswSchedulableEntity` (depending on the specified events for SWC resp. BSW). The `SwcBswRunnableMapping` instance controlling this case is only valid if several attributes of the mapped `RunnableEntity` and `BswSchedulableEntity` are consistent, especially all of the following constraints apply to the attributes of the given instance of `SwcBswRunnableMapping`:

- `swcRunnable`.`symbol` shall be identical to the symbol of `bswEntity` as defined in [TPS_BSWMDT_04138].

- `swcRunnable`.`minimumStartInterval` shall be identical to `bswEntity`.`minimumStartInterval`.

- `swcRunnable`.`canBeInvokedConcurrently` shall be identical to `bswEntity`.`implementedEntry`.`isReentrant`.

- `swcRunnable`.`swAddrMethod` shall either be empty or shall have identical attributes as the `SwAddrMethod` defined via `bswEntity`.`swAddrMethod`. This is required to ensure a unique configuration for the memory segment of the underlying code entity.

- `swcRunnable`.`activationReason` and `bswEntity`.`activationReason` shall have identical `shortName` if they define the same `bitPosition` and shall have identical `bitPosition` if they define the same `shortName`

Please note also the SWS_RTE for further details.

⌋*()*

[constr_4072] Constraints of `SectionNamePrefix.implementedIn` ⌈

- The `SectionNamePrefix` and the `DependencyOnArtifact` connected via this link shall belong to the same `BswImplementation`.

- The `DependencyOnArtifact` referred by this link shall be aggregated by `BswImplementation` in the role `requiredArtifact`.

- The `DependencyOnArtifact` referred by this link shall have the `category` value set to MEMMAP.

⌋*()*

[constr_4073] `McDataAccessDetails` shall refer to one ECU Extract ⌈Within one given `McDataAccessDetails`, all instances of `System` referenced as the base of any `McDataAccessDetails.variableAccess` or as the base of any `McDataAccessDetails.rteEvent` shall be identical and of `category` ECU_EXTRACT.

⌋*()*

[constr_4074] Compatibility of `BswModuleClientServerEntry`-s ⌈Two `BswModuleClientServerEntry`-s are compatible if and only if all of the following conditions hold:

- Their synchronicity values are identical. These values are taken from the attribute `isSynchronous` or, if this is undefined, from `encapsulatedEntry.isSynchronous`.

- The two `BswModuleEntry`-s referred as `encapsulatedEntry` have `SwServiceArg`, `returnType`, `serviceId` and `swServiceImplPolicy` identical.

⌋*()*

[constr_4075] Constraints for `providedData` and `requiredData` ⌈Sender-Receiver communication in BSW is restricted to the pattern of so-called *explicit communication* (in the same way as described for software components in [2]) with queued behavior. This leads to some constraints for the `VariableDataPrototype` referred in the role `BswModuleDescription.providedData` or `BswModuleDescription.requiredData`:

- It shall not have an `initValue`.

- Its `swDataDefProps.swImplPolicy` shall be set to `queued`.

- Its `swDataDefProps.swCalibrationAccess` shall be set to `notAccessible`.

There are no further formal constraints on the attributes of the `VariableDataPrototype` to be used in these roles or on the underlying `AutosarDataPrototype`.

⌋*()*

[constr_4076] Constraints on `BswModuleEntry` used for Client-Server ⌈A `BswModuleEntry` used in the role `BswModuleClientServerEntry.encapsulatedEntry` shall have attribute values as follows:

- `callType` shall be `regular` or `callback`.

- `executionContext` shall be `task`.

⌋*()*

[constr_4077] Constraints for `BswModuleEntity.reentrancyLevel` ⌈

- If the attribute `isReentrant` of a `BswModuleEntry` referred by an `BswModuleEntity` in the role `implementedEntry` has the value `True`, then the attribute `reentrancyLevel` of the same `BswModuleEntity` (if it exists) can only have the values `singleCoreReentrant` or `multicoreReentrant`.

- If the attribute `isReentrant` of a `BswModuleEntry` referred by an `BswModuleEntity` in the role `implementedEntry` has the values `False`, then there are no restrictions for the values of the attribute `reentrancyLevel` of the same `BswModuleEntity` (if it exists).

⌋*()*

[constr_4078] Consistent usage of `BswOperationInvokedEvent` ⌈The `BswCalledEntity` referred by the attribute `BswOperationInvokedEvent.startsOnEvent` shall refer to the same `BswModuleEntry` (via its attribute `implementedEntry`) as the `BswOperationInvokedEvent` (via its attribute `entry.encapsulatedEntry`.

⌋*()*

[constr_4079] `calledEntry` constraints for client-server calls ⌈

- The `BswModuleClientServerEntry` aggregated as `calledEntry` in a `BswSynchronousServerCallPoint` shall have the attribute `isSynchronous = True`.

- The `BswModuleClientServerEntry` aggregated as `calledEntry` in a `BswAsynchronousServerCallPoint` shall have the attribute `isSynchronous = false`.

⌋*()*

[constr_4080] Existence of reception policy ⌈If a `VariableDataPrototype` is referred from a `dataReceivePoint` of any `BswModuleEntity` in a given `BswInternalBehavior`, then exactly one corresponding `BswDataReceptionPolicy` shall by aggregated by this `BswInternalBehavior`.

⌋*()*

[constr_4081] Mode group used by BSW mode manager error event ⌈The `ModeDeclarationGroupPrototype` used by `BswModeManagerErrorEvent` shall be referred as `BswModuleDescription.providedModeGroup` by the same module.

⌋*()*

[constr_4083] `BswDistinguishedPartition` shall be used only in the context of a particular `BswInternalBehavior` ⌈All instances of `BswEvent`, `BswModuleCallPoint` and `BswVariableAccess` which refer to a `BswDistinguishedPar-`

`tition` shall belong to the same `BswInternalBehavior` that also aggregates the referred `BswDistinguishedPartition`.

⌋*()*

[constr_4084] Consistency of references of `InternalBehavior` ⌈The `SwcInternalBehavior` referenced by `SwcBswMapping.swcBehavior` in the `SwcBswMapping` determined by `SwcImplementation.swcBswMapping` shall be identical to the `SwcInternalBehavior` referenced by `SwcImplementation.behavior`.

⌋*()*

[constr_4085] Consistency of references of `InternalBehavior` ⌈The `BswInternalBehavior` referenced by `SwcBswMapping.bswBehavior` in the `SwcBswMapping` determined by `BswImplementation.swcBswMapping` shall be identical to the `BswInternalBehavior` referenced by `BswImplementation.behavior`.

⌋*()*

[constr_4086] invocation of `ExecutableEntity`s by direct function call dependent from `BswExecutionContext` ⌈The invocation of an `ExecutableEntity` with an `interruptCat1` can be implemented with a direct function call if the `BswExecutionContext` of the caller `BswModuleEntry` is set to `task`, `interruptCat2` or `interruptCat1`.

This applies to the invocation of a `triggered ExecutableEntity` by the `SchM_Trigger`, `SchM_ActMain` or `Rte_Trigger` APIs, or to the invocation of an `On Entry ExecutableEntity`, `OnTransition ExecutableEntity`, `OnExit ExecutableEntity` or `mode switch acknowledge ExecutableEntity` by the `SchM_Switch` or `Rte_Switch` APIs. For more information about the technical terms refer to [3]

⌋*()*

[constr_4087] Usage of category "MACRO" ⌈
It is only allowed to use the category "MACRO" for `SwServiceArg` if the owning `BswModuleEntry` has its `swServiceImplPolicy` attribute set to macro.

⌋*()*

[constr_4088] Existence of `RoleBasedDataTypeAssignment.role` vs. `RoleBasedDataAssignment.role` ⌈The usage of a `RoleBasedDataTypeAssignment` with attribute `role` set to the value `temporaryRamBlock` is only allowed if **no** `RoleBasedDataAssignment` defined with attribute `role` set to value `defaultValue` exists in the owning `BswServiceDependency`.

⌋*()*

[constr_4089] Association `callbackHeader` is only applicable for BSW modules ⌈The association `callbackHeader` is only supported for `codeDescriptor`s

of `BswImplementation` and only permitted to reference `ServiceNeeds` owned by `BswServiceDependency`.

⌋*()*

[constr_4090] The `callbackHeader` reference has to be consistent with behavior reference ⌈The reference `callbackHeader` is only allowed to reference `ServiceNeeds` in the context of the `BswServiceDependency` which in turn is referenced by the `BswImplementation` behavior of the `BswImplementation` owning the `codeDescriptor`.

⌋*()*

[constr_4091] `AccessCount.value` needs to be unambiguous ⌈AUTOSAR model shall define at most one `AccessCount.value` per `countProfile` for a specific `AbstractAccessPoint`.

⌋*()*

[constr_4092] Number of `ErrorTracerNeeds` in `BswInternalBehavior` ⌈A Bsw InternalBehavior shall provide at most one `ErrorTracerNeeds` element.

⌋*()*

[constr_4093] Entries linked to `BswModuleEntry`s shall have compatible signature ⌈Matching `BswModuleEntry`s according to [TPS_BSWMDT_04130] are compatible if the following condtions are fullfilled:

- both or neither of them define a `returnType`
- when the `returnType`s are defined, the `SwServiceArg`s in the role `returnType` shall be compatible
- both define the same number of compatible arguments in same order

⌋*()*

[constr_4094] compatibility of `SwServiceArg` in role `returnType` ⌈SwServiceArg in role `returnType` are compatible if they are identically typed

⌋*()*

[constr_4095] Compatibility of `SwServiceArg` in role `argument` ⌈SwServiceArg in role `returnType` are compatible if:

- they are identically typed

and

- if both do have the same `shortName`

⌋*()*

[constr_4096] Matching `BswModuleEntry`s should have compatible attributes ⌈Matching `BswModuleEntry`s according to [TPS_BSWMDT_04130] should be defined with identical values of the attributes

- `callType`

- `executionContext`

- `isReentrant`

- `isSynchronous`

- `serviceId`

- `swServiceImplPolicy`

- `bswEntryKind`

⌋*()*

[constr_4097] Limitation on the number of `BswExclusiveAreaPolicy`s ⌈An `ExclusiveArea` can only be referenced by at most one `BswExclusiveAreaPolicy`.

⌋*()*

[constr_4098] No mode disabling for `BswOperationInvokedEvent` ⌈A `BswOperationInvokedEvent` shall not have a reference to a `ModeDeclaration` in the role `disabledInMode`.

⌋*()*

[constr_4099] Support of multiple instantiation ⌈If a BSW Module supports multiple instantiation the attribute `vendorApiInfix` is mandatory.

⌋*()*

[constr_4100] Uniqueness of module implementation prefixes ⌈Inside one ECU the Module implementation prefixes (Mip) of BSW Modules shall be unique.

⌋*()*

[constr_4101] Semantics of `McGroupDataRefSet.flatMapEntry` ⌈

- An `McGroupDataRefSet` aggregated in the role of `McGroup.refCalprmSet` or `McGroup.refCalprmSet` shall only refer to `FlatInstanceDescriptor`s that can either be traced down to a `ParameterDataPrototype` or can be traced down to a `VariableDataPrototype` of category `COM_AXIS`, `RES_AXIS`, `CURVE`, `MAP`, `CUBOID`, `CUBE_4`, `CUBE_5` or `VAL_BLK` and which are declared for calibration access i.e. have an associated `SwDataDefProps`.`swCalibrationAccess` set to `readWrite` or `readOnly`.

- An `McGroupDataRefSet` aggregated in the role of `McGroup.refMeasurementSet` shall only refer to `FlatInstanceDescriptor`s that can be traced down to either a `VariableDataPrototype`, an `ArgumentDataPrototype` or a `ModeDeclarationGroupPrototype` and are declared as measurable i.e.

have an associated `SwDataDefProps.swCalibrationAccess` set to `read-Only`.

⌋*()*

[constr_4102] Semantics of `McGroupDataRefSet.mcDataInstance` ⌈

- An `McGroupDataRefSet` aggregated in the role of `McGroup.refCalprmSet` shall only refer to `McDataInstance`s that are declared for calibration access i.e. are aggregated in the role `McSupportData.mcParameterInstance` or `McSupportData.mcParameterInstance` of category `COM_AXIS`, `RES_AXIS`, `CURVE`, `MAP`, `CUBOID`, `CUBE_4`, `CUBE_5` or `VAL_BLK`.

- An `McGroupDataRefSet` aggregated in the role of `McGroup.refMeasurementSet` shall only refer to `McDataInstance`s that are declared as measurable i.e. are aggregated in the role `McSupportData.mcVariableInstance`.

⌋*()*

[constr_4103] Name convention for `SectionNamePrefix.symbol` ⌈In case a BSW module is split into allocatable memory parts the existing (according to [SWS_MemMap_00041]) `SectionNamePrefix.symbol` shall be set in the <MIP>_<FEATURE> form, where:

- <MIP> : is the capitalized module implementation prefix

- <FEATURE> : is the name of the sub-feature in the BSW module denoting the allocatable memory part

⌋*()*

[constr_4104] Referencing of `MemorySection`s to `SectionNamePrefix` ⌈In case a BSW module or Software Component is split into allocatable memory parts all `MemorySection`s belonging to the same allocatable memory part shall reference the identical `SectionNamePrefix` representing the allocatable memory part.

⌋*()*

[constr_4105] Use of attribute `task` or `cat2Isr` ⌈Only one of the attributes is allowed to exist. Either `task` or `cat2Isr` should be configured.

⌋*()*

[constr_4106] Restriction for the value of `SwServiceArg.swImplPolicy` ⌈The attribute `SwServiceArg.swImplPolicy` shall only have one of the following values:

- `SwImplPolicyEnum.const`

- `SwImplPolicyEnum.standard`

⌋*()*

[constr_4107] `swImplPolicy` for `SwServiceArg` ⌈The overriding value of attribute `swImplPolicy` of a `SwServiceArg` shall be `standard` or `const`. This rule

shall be imposed **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_4108] Restriction regarding the value of `SwServiceArg.category` ⌈The attribute `SwServiceArg.category` shall only have the following values:

- `VALUE`[2]

- `DATA_REFERENCE`

- `FUNCTION_REFERENCE`

- `TYPE_REFERENCE`

- `MACRO`

⌋*()*

[constr_10257] Existence of attribute `BswServiceDependency.serviceNeeds` ⌈For each `BswServiceDependency`, the attribute `serviceNeeds` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10258] Existence of the reference in the role `RoleBasedBswModuleEntryAssignment.assignedEntry` ⌈For each `RoleBasedBswModuleEntryAssignment`, the reference in the role `assignedEntry` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10259] Existence of attribute `RoleBasedBswModuleEntryAssignment.role` ⌈For each `RoleBasedBswModuleEntryAssignment`, the attribute `role` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10260] Existence of attribute `BswModuleEntry.callType` ⌈For each `BswModuleEntry`, the attribute `callType` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10261] Existence of attribute `BswModuleEntry.executionContext` ⌈For each `BswModuleEntry`, the attribute `executionContext` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

---

[2]This option has **very few** valid use cases, e.g. for defining a function pointer in native C notation, for example: `int (*SwCluC_BManif_VoidFncPtrType)(void);`

[constr_10262] Existence of attribute `BswModuleEntry.isReentrant` ⌈For each `BswModuleEntry`, the attribute `isReentrant` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10263] Existence of attribute `BswModuleEntry.isSynchronous` ⌈For each `BswModuleEntry`, the attribute `isSynchronous` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10264] Existence of attribute `BswModuleEntry.swServiceImplPolicy` ⌈For each `BswModuleEntry`, the attribute `swServiceImplPolicy` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10265] Existence of attribute `BswEntryRelationshipSet.bswEntryRelationship` ⌈For each `BswEntryRelationshipSet`, the attribute `bswEntryRelationship` shall exist at least once **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10266] Existence of attribute `BswEntryRelationship.bswEntryRelationshipType` ⌈For each `BswEntryRelationship`, the attribute `bswEntryRelationshipType` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10267] Existence of reference in the role `BswEntryRelationship.from` ⌈For each `BswEntryRelationship`, the reference in the role `from` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10268] Existence of reference in the role `BswEntryRelationship.to` ⌈For each `BswEntryRelationship`, the reference in the role `to` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10269] Existence of the reference in the role `BswModuleClientServerEntry.encapsulatedEntry` ⌈For each `BswModuleClientServerEntry`, the the reference in the role `encapsulatedEntry` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10270] Existence of attribute `AccessCountSet.countProfile` ⌈For each `AccessCountSet`, the attribute `countProfile` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10271] Existence of attribute `AccessCount.value` ⌈For each `AccessCount`, the attribute `value` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10272] Existence of the reference in the role `BswModuleEntity.implementedEntry` ⌈For each `BswModuleEntity`, the reference in the role `implementedEntry` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10273] Existence of attribute `BswInterruptEntity.interruptCategory` ⌈For each `BswInterruptEntity`, the attribute `interruptCategory` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10274] Existence of attribute `BswInterruptEntity.interruptSource` ⌈For each `BswInterruptEntity`, the attribute `interruptSource` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10275] Existence of the reference in the role `BswDirectCallPoint.calledEntry` ⌈For each `BswDirectCallPoint`, the reference in the role `calledEntry` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10276] Existence of the reference in the role `BswSynchronousServerCallPoint.calledEntry` ⌈For each `BswSynchronousServerCallPoint`, the reference in the role `calledEntry` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10277] Existence of the reference in the role `BswAsynchronousServerCallPoint.calledEntry` ⌈For each `BswAsynchronousServerCallPoint`, the reference in the role `calledEntry` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10278] Existence of the reference in the role `BswAsynchronousServer-CallResultPoint.asynchronousServerCallPoint` ⌈For each `BswAsynchronousServerCallResultPoint`, the reference in the role `asynchronousServerCallPoint` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10279] Existence of the reference in the role `BswVariableAccess.accessedVariable` ⌈For each `BswVariableAccess`, the reference in the role `accessedVariable` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10280] Existence of the reference in the role `BswExclusiveAreaPolicy.exclusiveArea` ⌈For each `BswExclusiveAreaPolicy`, the reference in the role `exclusiveArea` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10281] Existence of attribute `BswTimingEvent.period` ⌈For each `BswTimingEvent`, the attribute `period` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10282] Existence of the reference in the role `BswInternalTriggerOccurredEvent.eventSource` ⌈For each `BswInternalTriggerOccurredEvent`, the reference in the role `eventSource` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10283] Existence of the reference in the role `BswExternalTriggerOccurredEvent.trigger` ⌈For each `BswExternalTriggerOccurredEvent`, the reference in the role `trigger` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10284] Existence of attribute `BswModeSwitchEvent.activation` ⌈For each `BswModeSwitchEvent`, the attribute `activation` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10285] Existence of the reference in the role `BswModeSwitchedAckEvent.modeGroup` ⌈For each `BswModeSwitchedAckEvent`, the reference in the

role `modeGroup` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10286] Existence of the reference in the role `BswModeManagerErrorEvent.modeGroup` ⌈For each `BswModeManagerErrorEvent`, the reference in the role `modeGroup` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10287] Existence of the reference in the role `BswOperationInvokedEvent.entry` ⌈For each `BswOperationInvokedEvent`, the reference in the role `entry` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10288] Existence of the reference in the role `BswAsynchronousServerCallReturnsEvent.eventSource` ⌈For each `BswAsynchronousServerCallReturnsEvent`, the reference in the role `eventSource` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10289] Existence of the reference in the role `BswDataReceivedEvent.data` ⌈For each `BswDataReceivedEvent`, the reference in the role `data` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10290] Existence of the reference in the role `BswTriggerDirectImplementation.masteredTrigger` ⌈For each `BswTriggerDirectImplementation`, the reference in the role `masteredTrigger` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10291] Existence of the reference in the role `BswModeSenderPolicy.providedModeGroup` ⌈For each `BswModeSenderPolicy`, the reference in the role `providedModeGroup` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10292] Existence of attribute `BswModeSenderPolicy.queueLength` ⌈For each `BswModeSenderPolicy`, the attribute `queueLength` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10293] Existence of attribute `BswModeSwitchAckRequest.timeout` ⌈For each `BswModeSwitchAckRequest`, the attribute `timeout` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10294] Existence of the reference in the role `BswModeReceiverPolicy`. `requiredModeGroup` ⌈For each `BswModeReceiverPolicy`, the reference in the role `requiredModeGroup` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10295] Existence of attribute `BswModeReceiverPolicy.supportsAsynchronousModeSwitch` ⌈For each `BswModeReceiverPolicy`, the attribute `supportsAsynchronousModeSwitch` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10296] Existence of reference in the role `BswDataReceptionPolicy`. `receivedData` ⌈For each `BswDataReceptionPolicy`, the reference in the role `receivedData` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10297] Existence of attribute `BswQueuedDataReceptionPolicy`. `queueLength` ⌈For each `BswQueuedDataReceptionPolicy`, the attribute `queueLength` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10298] Existence of the reference in the role `SwcBswRunnableMapping`. `bswEntity` ⌈For each `SwcBswRunnableMapping`, the reference in the role `bswEntity` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10299] Existence of the reference in the role `SwcBswRunnableMapping.swcRunnable` ⌈For each `SwcBswRunnableMapping`, the reference in the role `swcRunnable` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10300] Existence of the reference in the role `SwcBswSynchronizedTrigger.bswTrigger` ⌈For each `SwcBswSynchronizedTrigger`, the ref-

erence in the role `bswTrigger` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10301] **Existence of the instanceRef in the role `SwcBswSynchronizedTrigger.swcTrigger`** ⌈For each `SwcBswSynchronizedTrigger`, the instanceRef in the role `swcTrigger` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10302] Existence of attribute `BswImplementation.arReleaseVersion` ⌈For each `BswImplementation`, the attribute `arReleaseVersion` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10303] **Existence of the reference in the role `BswImplementation.behavior`** ⌈For each `BswImplementation`, the reference in the role `behavior` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10304] Existence of attribute `DependencyOnArtifact.usage` ⌈For each `DependencyOnArtifact`, the attribute `usage` shall exist at least once **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10305] **Existence of attribute `WorstCaseStackUsage.memoryConsumption`** ⌈For each `WorstCaseStackUsage`, the attribute `memoryConsumption` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10306] **Existence of attribute `MeasuredStackUsage.averageMemoryConsumption`** ⌈For each `MeasuredStackUsage`, the attribute `averageMemoryConsumption` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10307] **Existence of attribute `MeasuredStackUsage.maximumMemoryConsumption`** ⌈For each `MeasuredStackUsage`, the attribute `maximumMemoryConsumption` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10308] Existence of attribute `RoughEstimateStackUsage.memoryConsumption` ⌈For each `RoughEstimateStackUsage`, the attribute `memoryConsumption` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10309] Existence of attribute `WorstCaseHeapUsage.memoryConsumption` ⌈For each `WorstCaseHeapUsage`, the attribute `memoryConsumption` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10310] Existence of attribute `MeasuredHeapUsage.averageMemoryConsumption` ⌈For each `MeasuredHeapUsage`, the attribute `averageMemoryConsumption` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10311] Existence of attribute `MeasuredHeapUsage.maximumMemoryConsumption` ⌈For each `MeasuredHeapUsage`, the attribute `maximumMemoryConsumption` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10312] Existence of attribute `RoughEstimateHeapUsage.memoryConsumption` ⌈For each `RoughEstimateHeapUsage`, the attribute `memoryConsumption` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10313] Existence of attribute `ExecutionTime.hardwareConfiguration` ⌈For each `ExecutionTime`, the attribute `hardwareConfiguration` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10314] Existence of attribute `ExecutionTime.softwareContext` ⌈For each `ExecutionTime`, the attribute `softwareContext` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10315] Existence of attribute `HardwareConfiguration.additionalInformation` ⌈For each `HardwareConfiguration`, the attribute `additionalIn-

`formation` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10316]   Existence of attribute `HardwareConfiguration.processorMode` ⌈For each `HardwareConfiguration`, the attribute `processorMode` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10317] Existence of attribute `HardwareConfiguration.processorSpeed` ⌈For each `HardwareConfiguration`, the attribute `processorSpeed` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10318] Existence of reference `MemorySectionLocation.providedMemory` ⌈For each `MemorySectionLocation`, the reference in the role `providedMemory` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10319] Existence of reference `MemorySectionLocation.softwareMemorySection` ⌈For each `MemorySectionLocation`, the reference in the role `softwareMemorySection` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10320] Existence of attribute `SoftwareContext.input` ⌈For each `SoftwareContext`, the attribute `input` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10321] Existence of attribute `SoftwareContext.state` ⌈For each `SoftwareContext`, the attribute `state` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10323] Existence of attribute `AnalyzedExecutionTime.bestCaseExecutionTime` ⌈For each `AnalyzedExecutionTime`, the attribute `bestCaseExecutionTime` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10324] Existence of attribute `AnalyzedExecutionTime.worstCaseExecutionTime` ⌈For each `AnalyzedExecutionTime`, the attribute `worstCaseExecutionTime` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10325] Existence of attribute `MeasuredExecutionTime.maximumExecutionTime` ⌈For each `MeasuredExecutionTime`, the attribute `maximumExecutionTime` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10326] Existence of attribute `MeasuredExecutionTime.minimumExecutionTime` ⌈For each `MeasuredExecutionTime`, the attribute `minimumExecutionTime` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10327] Existence of attribute `MeasuredExecutionTime.nominalExecutionTime` ⌈For each `MeasuredExecutionTime`, the attribute `nominalExecutionTime` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10328] Existence of the reference in the role `BswEvent.startsOnEvent` ⌈For each `BswEvent`, the reference in the role `startsOnEvent` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10329] Existence of the instanceRef in the role `McDataAccessDetails.variableAccess` ⌈For each `McDataAccessDetails`, the instanceRef in the role `variableAccess` shall exist at least once **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10330] Existence of attribute `RptServicePoint.symbol` ⌈For each `RptServicePoint`, the attribute `symbol` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10331] Existence of attribute `SimulatedExecutionTime.maximumExecutionTime` ⌈For each `SimulatedExecutionTime`, the attribute `maximumExecutionTime` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10332] Existence of attribute `SimulatedExecutionTime.minimumExecutionTime` ⌈For each `SimulatedExecutionTime`, the attribute `minimumExecutionTime` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10333] Existence of attribute `SimulatedExecutionTime.nominalExecutionTime` ⌈For each `SimulatedExecutionTime`, the attribute `nominalExecutionTime` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10334] Existence of attribute `RoughEstimateOfExecutionTime.additionalInformation` ⌈For each `RoughEstimateOfExecutionTime`, the attribute `additionalInformation` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10335] Existence of attribute `RoughEstimateOfExecutionTime.estimatedExecutionTime` ⌈For each `RoughEstimateOfExecutionTime`, the attribute `estimatedExecutionTime` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10336] Existence of the reference in the role `SwcBswSynchronizedModeGroupPrototype.bswModeGroup` ⌈For each `SwcBswSynchronizedModeGroupPrototype`, the reference in the role `bswModeGroup` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10337] Existence of the instanceRef in the role `SwcBswSynchronizedModeGroupPrototype.swcModeGroup` ⌈For each `SwcBswSynchronizedModeGroupPrototype`, the instanceRef in the role `swcModeGroup` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10338] Existence of attribute `MultidimensionalTime.cseCode` ⌈For each `MultidimensionalTime`, the attribute `cseCode` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10339] Existence of attribute `MultidimensionalTime.cseCodeFactor` ⌈For each `MultidimensionalTime`, the attribute `cseCodeFactor` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10340]   Existence of attribute `McSwEmulationMethodSupport.category` ⌈For each `McSwEmulationMethodSupport`, the attribute `category` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10341] Existence of attribute `McSwEmulationMethodSupport.shortLabel` ⌈For each `McSwEmulationMethodSupport`, the attribute `shortLabel` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10342] Existence of the reference in the role `McParameterElementGroup.ramLocation` ⌈For each `McParameterElementGroup`, the reference in the role `ramLocation` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10343] Existence of the reference in the role `McParameterElementGroup.romLocation` ⌈For each `McParameterElementGroup`, the reference in the role `romLocation` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10344] Existence of attribute `McParameterElementGroup.shortLabel` ⌈For each `McParameterElementGroup`, the attribute `shortLabel` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10345]   Existence of the reference in the role `ImplementationElementInParameterInstanceRef.context` ⌈For each `ImplementationElementInParameterInstanceRef`, the reference in the role `context` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10346]   Existence of the reference in the role `ImplementationElementInParameterInstanceRef.target` ⌈For each `ImplementationElementInParameterInstanceRef`, the reference in the role `target` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10347] Existence of the instanceRef in the role `McDataAccessDetails.rteEvent` ⌈For each `McDataAccessDetails`, the instanceRef in the role `rteEvent`

shall exist at least once **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10349] Existence of attribute `RptSupportData.executionContext` ⌈For each `RptSupportData`, the attribute `executionContext` shall exist at least once **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10350] Existence of attribute `RptSupportData.rptComponent` ⌈For each `RptSupportData`, the attribute `rptComponent` shall exist at least once **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10351] Existence of attribute `RptSupportData.rptServicePoint` ⌈For each `RptSupportData`, the attribute `rptServicePoint` shall exist at least once **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10352] Existence of attribute `RptComponent.rptExecutableEntity` ⌈For each `RptComponent`, the attribute `rptExecutableEntity` shall exist at least once **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10353] Existence of attribute `RptExecutableEntity.rptExecutableEntityEvent` ⌈For each `RptExecutableEntity`, the attribute `rptExecutableEntityEvent` shall exist at least once **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10354] Existence of attribute `RptExecutableEntity.symbol` ⌈For each `RptExecutableEntity`, the attribute `symbol` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10355] Existence of the reference in the role `RptExecutableEntityEvent.executionContext` ⌈For each `RptExecutableEntityEvent`, the reference in the role `executionContext` shall exist at least once **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10356] Existence of attribute `RptExecutableEntityEvent.rptEventId` ⌈For each `RptExecutableEntityEvent`, the attribute `rptEventId` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10357] Existence of attribute `RptExecutableEntityEvent.rptExecutableEntityProperties` ⌈For each `RptExecutableEntityEvent`, the attribute `rptExecutableEntityProperties` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10358] Existence of the reference in the role `RptExecutableEntityEvent.rptServicePointPost` ⌈For each `RptExecutableEntityEvent`, the reference in the role `rptServicePointPost` shall exist at least once **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10359] Existence of the reference in the role `RptExecutableEntityEvent.rptServicePointPre` ⌈For each `RptExecutableEntityEvent`, the reference in the role `rptServicePointPre` shall exist at least once **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10360] Existence of attribute `RptServicePoint.serviceId` ⌈For each `RptServicePoint`, the attribute `serviceId` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10362] Existence of attribute `AliasNameSet.aliasName` ⌈For each `AliasNameSet`, the attribute `aliasName` shall exist at least once **at the time when the configuration of the BSW module is finished**.

⌋*()*

[constr_10363] Existence of attribute `AliasNameAssignment.shortLabel` ⌈For each `AliasNameAssignment`, the attribute `shortLabel` shall exist **at the time when the configuration of the BSW module is finished**.

⌋*()*

2.2 TPS_DiagnosticExtractTemplate

[constr_1324] Existence of attribute `DiagnosticDataIdentifier.representsVin` ⌈Within the context of a given `DiagnosticContributionSet`, the attribute `DiagnosticDataIdentifier.representsVin` shall have the value `true` for only a single `DiagnosticDataIdentifier` **at the time when the DEXT is complete**.

⌋*()*

[constr_1325] Allowed attributes of `SwDataDefProps` for `DiagnosticDataElement.swDataDefProps` ⌈

Attributes of `SwDataDefProps`	`DiagnosticDataElement.swDataDefProps`
`additionalNativeTypeQualifier`	N/A
`annotation`	N/A
`baseType.baseTypeDefinition.baseTypeEncoding`	D
`baseType.baseTypeDefinition.baseTypeSize`	D
`baseType.baseTypeDefinition.byteOrder`	D
`baseType.baseTypeDefinition.memAlignment`	N/A
`baseType.baseTypeDefinition.nativeDeclaration`	N/A
`compuMethod`	D
`dataConstr`	D
`displayFormat`	D
`displayPresentation`	N/A
`implementationDataType`	N/A
`invalidValue`	N/A
`swAddrMethod`	N/A
`swAlignment`	N/A
`swBitRepresentation`	N/A
`swCalibrationAccess`	N/A
`swCalprmAxisSet`	N/A
`swComparisonVariable`	N/A
`swDataDependency`	N/A
`swImplPolicy`	N/A
`swIntendedResolution`	N/A
`swInterpolationMethod`	N/A
`swIsVirtual`	N/A
`swPointerTargetProps`	N/A
`swRecordLayout`	N/A
`swRefreshTiming`	N/A
`swTextProps`	N/A
`swValueBlockSize`	N/A
`unit`	D
`valueAxisDataType`	N/A

⌋*()*

[constr_1326] Existence of a variable-sized array ⌈The value of the attribute `DiagnosticDataElement.arraySizeSemantics` **shall not** be set to `ArraySizeSemanticsEnum.variableSize` if the respective `DiagnosticDataElement` is referenced from a `DiagnosticServiceDataMapping`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1327]   Multiplicity of `DiagnosticEcuInstanceProps.ecuInstance` ⌈The multiplicity of `DiagnosticEcuInstanceProps.ecuInstance` shall be limited to 1 and the enclosing `DiagnosticContributionSet` shall only refer to at most one `DiagnosticEcuInstanceProps` **if** the enclosing `DiagnosticContributionSet` is of `category` DIAGNOSTICS_ECU_EXTRACT.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1328] **Consistency of `DiagnosticEcuInstanceProps.ecuInstance`
and `DiagnosticServiceTable.ecuInstance`** ⌈Each `DiagnosticServic-
eTable` referenced by any given `DiagnosticContributionSet` in the role
`serviceTable` shall define a reference in the role `DiagnosticServiceTable.
ecuInstance` to an `EcuInstance` that is also referenced in the role `Diagnos-
ticEcuInstanceProps.ecuInstance` by a `DiagnosticEcuInstanceProps`
referenced by the mentioned `DiagnosticContributionSet` **if** the respective
`DiagnosticContributionSet` is of `category` DIAGNOSTICS_ECU_EXTRACT.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

**[constr_1329] Existence of concrete sub-classes of `DiagnosticServiceClass`
in the context created by a `DiagnosticContributionSet`** ⌈One of the following
mutually exclusive conditions shall apply for the existence of any concrete sub-class
of `DiagnosticServiceClass` in the context created by a `DiagnosticContribu-
tionSet`:

- The subclass of `DiagnosticServiceClass` shall only appear once in the con-
  text created by a `DiagnosticContributionSet`

- If the subclass of `DiagnosticServiceClass` appears multiple times in the
  context created by a `DiagnosticContributionSet` then all instances shall
  have identical values for all of their attributes.

  In case of aggregations the number of aggregated elements shall be identical and
  the values of primitive attributes of aggregated elements shall again be identical.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

**[constr_1330] Custom service identifier shall not overlap with standardized ser-
vice identifiers** ⌈The value of the attribute `customServiceId` shall not be set to
any of the values reserved for standardized service identifiers as defined by the ISO
14229-1, see [4].

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1331] Existence of `DiagnosticEcuReset.customSubFunctionNumber`
⌈The attribute `DiagnosticEcuReset.customSubFunctionNumber` shall only exist
if the value of `DiagnosticEcuReset.category` is outside the standardized set of
values as defined by [TPS_DEXT_01056].

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1332] Value range for `DiagnosticEcuReset.customSubFunctionNumber` ⌈The allowed value for `DiagnosticEcuReset.customSubFunctionNumber` shall always be within the closed interval **0x40** .. **0x7E**.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1333] Existence of `DiagnosticMemoryIdentifier.memoryLowAddress` and `DiagnosticMemoryIdentifier.memoryHighAddress` ⌈The attributes `DiagnosticMemoryIdentifier.memoryLowAddress` as well as `DiagnosticMemoryIdentifier.memoryHighAddress` shall not exist if the `DiagnosticMemoryIdentifier` referenced in the role `memoryRange` is referenced by a `DiagnosticRequestDownload` or a `DiagnosticRequestUpload`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1334] Existence of `DiagnosticComControl.customSubFunctionNumber` ⌈The attribute `DiagnosticComControl.customSubFunctionNumber` shall only exist if the value of `DiagnosticComControl.category` is outside the standardized set of values as defined by [TPS_DEXT_01057].

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1335] Possible values for `DiagnosticComControl.customSubFunctionNumber` ⌈Given the fulfillment of [constr_1334], the value of a given `DiagnosticComControl.customSubFunctionNumber` shall always be within the closed interval **0x40** .. **0x5F** (for manufacturer-specific sub-functions) or the closed interval **0x60** .. **0x7E** (for supplier-specific sub-functions).

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1336] Applicable value range for `DiagnosticComControlSpecificChannel.subnetNumber` ⌈The value of attribute `DiagnosticComControlSpecificChannel.subnetNumber` shall be within the closed interval **1** .. **14**.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1337] Allowed value range for attribute `DiagnosticComControlSubNodeChannel.subNodeNumber` ⌈The value of attribute `DiagnosticComControlSubNodeChannel.subNodeNumber` shall not exceed the closed interval **0** .. **65535**.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1338] Maximum number of aggregated `DiagnosticReadDataByPeriodicIDClass.periodicRate` ⌈The number of aggregated `periodicRate` within the context of one `DiagnosticReadDataByPeriodicIDClass` shall be within the closed interval 1..3.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1339] Existence of `DiagnosticRoutine.start` ⌈In a complete `DiagnosticExtract`, the attribute `DiagnosticRoutine.start` shall always exist for any given `DiagnosticRoutine`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1340] Consistency of `DiagnosticServiceSwMapping` with respect to synchronously called `DiagnosticRoutine`s ⌈Each `DiagnosticServiceSwMapping` that references a `DiagnosticRoutineControl` that only aggregates a `DiagnosticStartRoutine` in the role `start` shall only reference a `SwcServiceDependency` or `BswServiceDependency` that in turn aggregates a `DiagnosticRoutineNeeds` with attribute `diagRoutineType` set to `DiagnosticRoutineTypeEnum.synchronous`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1341] Consistency of `DiagnosticServiceSwMapping` with respect to asynchronously called `DiagnosticRoutine`s ⌈Each `DiagnosticServiceSwMapping` that references a `DiagnosticRoutineControl` that aggregates a `DiagnosticStopRoutine` and/or `DiagnosticRequestRoutineResults` in the role `stop` or `requestResult` shall only reference a `SwcServiceDependency` or `BswServiceDependency` that in turn aggregates a `DiagnosticRoutineNeeds` with attribute `diagRoutineType` set to `DiagnosticRoutineTypeEnum.asynchronous`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1342] Possible values for `DiagnosticSecurityAccess.requestSeedId` ⌈The value of the attribute `DiagnosticSecurityAccess.requestSeedId` **shall only be set to an odd number**[3].

The supported value range consists of the following list:

---

[3]The even numbers are reserved for the identification of the corresponding *sendKey* sub-function, as explained by [TPS_DEXT_01036].

- all odd numbers in the closed interval **0x01** .. **0x41**

- **0x5F** (this corresponds to the case of *end-of-life activation of on-board pyrotechnic devices according to ISO 26021-2 [5]*)

- all odd numbers in the closed interval **0x61** .. **0x7E**

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1343] Simultaneous existence of the attributes `DiagnosticService-DataMapping.diagnosticDataElement` and `DiagnosticDataByIdentifier.dataIdentifier` ⌈A `DiagnosticServiceDataMapping.diagnosticDataElement` shall also be aggregated by a `DiagnosticDataByIdentifier` in the role `dataIdentifier.dataElement.dataElement`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1345] `DiagnosticDataElement` shall not (finally) be aggregated by a `DiagnosticRoutine` ⌈A `DiagnosticDataElement` that is referenced by a `DiagnosticServiceDataMapping` shall not (finally) be aggregated by a `DiagnosticRoutine`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1346] Allowed values of `DiagnosticServiceSwMapping.serviceInstance` ⌈The applicability of the `DiagnosticServiceSwMapping` is limited to predefined set of diagnostic services.

By regulation of the AUTOSAR standard, `DiagnosticServiceSwMapping.serviceInstance` shall only point to the following sub-classes of `DiagnosticServiceInstance`:

- `DiagnosticRoutine`

- `DiagnosticSecurityAccess`

- `DiagnosticReadDataByIdentifier`

- `DiagnosticWriteDataByIdentifier`

- `DiagnosticIOControl`

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1347] Existence of attributes of `DiagnosticServiceSwMapping` ⌈For any given `DiagnosticServiceSwMapping`, **one and only one of** the following references shall exist:

- `DiagnosticServiceSwMapping.mappedFlatSwcServiceDependency`

- `DiagnosticServiceSwMapping.mappedSwcServiceDependencyInSystem`

- `DiagnosticServiceSwMapping.mappedBswServiceDependency`

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1349] Value of `udsDtcValue` shall be unique ⌈The value of `DiagnosticTroubleCodeUds.udsDtcValue` shall be unique for all `DiagnosticTroubleCodeUds` that refer to the same `DiagnosticMemoryDestination` via the reference `DiagnosticTroubleCodeUds.dtcProps.diagnosticMemory`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1350] Value of `DiagnosticTroubleCodeGroup.groupNumber` shall be unique ⌈The value of `DiagnosticTroubleCodeGroup.groupNumber` shall be unique to any other DTC and DTC group value.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1351] Value of `DiagnosticTroubleCodeGroup.groupNumber` ⌈To be compliant to ISO, the value of `DiagnosticTroubleCodeGroup.groupNumber` shall be set as defined in ISO 14229-1 [4].

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1352] Existence of `maxNumberFreezeFrameRecords` vs. `freezeFrame` ⌈If the attribute `DiagnosticTroubleCodeProps.maxNumberFreezeFrameRecords` exists than the attribute `DiagnosticTroubleCodeProps.freezeFrame` shall not exist or vice versa.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1353] Applicability of [constr_1352] ⌈[constr_1352] shall apply in the identical way (either one or the other attribute shall exist) for all `DiagnosticTroubleCodeProps` within the context of all `DiagnosticContributionSet`s of category `DIAGNOSTIC_ECU_EXTRACT` that refer to the same `EcuInstance`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1354] **Existence of attribute `DiagnosticTroubleCodeProps.snapshotRecordContent`** ⌈If one of the attributes `DiagnosticTroubleCodeProps.maxNumberFreezeFrameRecords` or `DiagnosticTroubleCodeProps.freezeFrame` exists then the attribute `DiagnosticTroubleCodeProps.snapshotRecordContent` shall exist.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1355] Value of `extendedDataRecord.recordNumber` ⌈To be compliant to ISO, the value of `extendedDataRecord.recordNumber` shall be set in the interval as defined in ISO 14229-1 [4].

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1357] Value of `freezeFrame.recordNumber` ⌈To be compliant to ISO, the value of `freezeFrame.recordNumber` shall be set in the interval as defined in ISO 14229-1 [4].

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1359] Condition for the existence of attribute `DiagnosticDebounceAlgorithmProps.debounceCounterStorage` ⌈Attribute `debounceCounterStorage` of meta-class `DiagnosticDebounceAlgorithmProps` shall only exist if the aggregation of attribute `debounceAlgorithm` at `DiagnosticDebounceAlgorithmProps` actually aggregates a `DiagEventDebounceCounterBased`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1361] **Number of `DiagnosticEventToEnableConditionGroupMapping` elements per `DiagnosticEvent`** ⌈The mapping element `DiagnosticEventToEnableConditionGroupMapping` shall be created no more than once per `DiagnosticEvent`.

If several `DiagnosticEventToEnableConditionGroupMapping` elements referring to the same `DiagnosticEvent` are defined, then the `Enable Condition Group` mapping shall be regarded as defective.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1362] **Number of `DiagnosticEventToStorageConditionGroupMapping` elements per `DiagnosticEvent`** ⌈The mapping element `DiagnosticEventToStorageConditionGroupMapping` shall be created no more than once or once per `DiagnosticEvent`.

If several `DiagnosticEventToStorageConditionGroupMapping` elements referring to the same `DiagnosticEvent` are defined, then the `Storage Condition Group` mapping shall be regarded as defective.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1378] Value of `DiagnosticMemoryDestinationUserDefined.memoryId` ⌈Within the scope of one `DiagnosticContributionSet`, no two (or more) `DiagnosticMemoryDestinationUserDefined`s shall exist that share the same value for attribute `DiagnosticMemoryDestinationUserDefined.memoryId`

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1379] Existence of `DiagnosticMemoryDestinationPrimary` ⌈Within the scope of one `DiagnosticContributionSet` only one `DiagnosticMemoryDestinationPrimary` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1394] Value of `DiagnosticDataElement.maxNumberOfElements` depending on its existence ⌈If the attribute `DiagnosticDataElement.maxNumberOfElements` exists then its value shall be greater than 0 **at the time when the DEXT is complete**.

⌋*()*

[constr_1405] Value of `DiagnosticProtocol.serviceTable` vs. `DiagnosticServiceTable.protocolKind` ⌈If the reference `DiagnosticProtocol.serviceTable` exists then the value of `DiagnosticProtocol.protocolKind` shall be identical to the value of `DiagnosticServiceTable.protocolKind`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1406] `DiagnosticServiceTable.diagnosticConnection` vs. `DiagnosticProtocol.diagnosticConnection` ⌈If a `DiagnosticServiceTable` exists that fulfills the following conditions:

- reference `DiagnosticServiceTable.diagnosticConnection` exists

- the `DiagnosticServiceTable` is referenced by means of `DiagnosticProtocol.serviceTable`

then all of the `DiagnosticConnection`s referenced by means of `DiagnosticServiceTable.diagnosticConnection` shall also be referenced in the role `diagnosticConnection` from a `DiagnosticProtocol` that in turn references the respective `DiagnosticServiceTable` in the role `DiagnosticProtocol.serviceTable`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1411] Existence of `DiagnosticMemoryIdentifier.memoryHighAddressLabel` vs. `DiagnosticMemoryIdentifier.memoryHighAddress` ⌈At most **one** of the attributes in the following list shall exist **at the time when the DEXT is complete**:

- `DiagnosticMemoryIdentifier.memoryHighAddressLabel`

- `DiagnosticMemoryIdentifier.memoryHighAddress`

⌋*()*

[constr_1412] Existence of `DiagnosticMemoryIdentifier.memoryLowAddressLabel` vs. `DiagnosticMemoryIdentifier.memoryLowAddress` ⌈At most **one** of the attributes in the following list shall exist **at the time when the DEXT is complete**:

- `DiagnosticMemoryIdentifier.memoryLowAddressLabel`

- `DiagnosticMemoryIdentifier.memoryLowAddress`

⌋*()*

[constr_1419] Value of `DiagnosticSecurityLevel.accessDataRecordSize` ⌈If the attribute `DiagnosticSecurityLevel.accessDataRecordSize` exists then its value shall be greater than zero **at the time when the DEXT is complete**.

⌋*()*

[constr_1421] Consistency of `DiagnosticDynamicallyDefineDataIdentifierClass.subfunction` ⌈The values of `DiagnosticDynamicallyDefineDataIdentifierClass.subfunction` shall not repeat, i.e. every value of `DiagnosticDynamicallyDefineDataIdentifierSubfunctionEnum` shall at most appear once in the `subfunction` attribute.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1435] Debouncing in the presence of a `DiagnosticEventPortMapping` ⌈If a `DiagnosticEventPortMapping` exists and the enclosed `DiagnosticEventPortMapping.diagnosticEvent` is also referenced by a `DiagnosticEventToDebounceAlgorithmMapping` then the concrete subclass of the respective `DiagnosticEventToDebounceAlgorithmMapping.debounceAlgorithm.debounceAlgorithm` shall be identical to the `DiagnosticEventPortMapping.swcServiceDependencyInSystem`/`swcFlatServiceDependency.serviceNeeds.diagEventDebounceAlgorithm`.

It is assumed that the target of reference `DiagnosticEventPortMapping`.`swcServiceDependencyInSystem` resp. `swcFlatServiceDependency` aggregates a `DiagnosticEventNeeds`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1447] Restrictions for the value of `DiagnosticParameterIdentifier.id` ⌈The values 0x00, 0x20, 0x40, 0x60, 0x80, 0xA0, 0xC0, and 0xE0 are not allowed to appear in the value of `DiagnosticParameterIdentifier`.id.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1448] Interval of `DiagnosticParameterIdentifier.id` ⌈The allowed interval for values of `DiagnosticParameterIdentifier`.id shall not exceed [0..255] **at the time when the DEXT is complete**.

⌋*()*

[constr_1449] PID shall only carry a fixed-length collection of data ⌈The value of `DiagnosticParameterIdentifier`.`dataElement`.`dataElement`.`arraySizeSemantics` shall not be set to `variableSize`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1450] Service mapping for ODB mode 0x01 for `DiagnosticParameterIdentifier` ⌈if a `DiagnosticServiceSwMapping` or `DiagnosticServiceDataMapping` refers to a `DiagnosticRequestCurrentPowertrainData` and a `DiagnosticDataElement` that is aggregated by a `DiagnosticParameterIdentifier` then then one of two alternative model configurations shall exist:

- `SwcServiceDependency` referenced by the same `DiagnosticServiceSwMapping` or `DiagnosticServiceDataMapping` shall aggregate an `ObdPidServiceNeeds` in the role `serviceNeeds`.

- The `BswServiceDependencyIdent` referenced by the same `DiagnosticServiceSwMapping` shall aggregate an `ObdPidServiceNeeds` in the role `serviceNeeds`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1451] Service mapping for OBD mode 0x09 for `DiagnosticInfoType` ⌈if a `DiagnosticServiceSwMapping` refers to `DiagnosticRequestVehicleInfo` and a `DiagnosticDataElement` that is aggregated by a `DiagnosticInfoType` then one of two alternative model configurations shall exist:

- The `SwcServiceDependency` referenced by the same `DiagnosticServiceSwMapping` shall aggregate a `ObdInfoServiceNeeds` in the role `serviceNeeds`.

- The `BswServiceDependencyIdent` referenced by the same `DiagnosticServiceSwMapping` shall aggregate an `ObdInfoServiceNeeds` in the role `serviceNeeds`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1452] Service mapping for OBD mode 0x08 for `DiagnosticInfoType`
⌈if a `DiagnosticServiceSwMapping` refers to a `DiagnosticRequestControlOfOnBoardDevice` then the `SwcServiceDependency` referenced by the same `DiagnosticServiceSwMapping` shall aggregate an `ObdControlServiceNeeds` in the role `serviceNeeds`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1453] References from `DiagnosticFunctionInhibitSource` ⌈Each `DiagnosticFunctionInhibitSource` may either reference one of the following meta-classes in their respective roles:

- `DiagnosticFimAliasEventMapping` in the role `event`

- `DiagnosticFimAliasEventGroupMapping` in the role `eventGroup`

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1454] `DiagnosticFimFunctionMapping` shall only reference a `SwcServiceDependency` that aggregates `FunctionInhibitionNeeds` ⌈A `DiagnosticFimFunctionMapping` shall only reference a `SwcServiceDependency` that aggregates `FunctionInhibitionNeeds` in the role `serviceNeeds`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1455] Relation of `DiagnosticJ1939Node` to `J1939NmNode` ⌈Each `J1939NmNode` shall only be referenced in the role `nmNode` by a single `DiagnosticJ1939Node`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1456] Valid interval for attribute `DiagnosticTroubleCodeJ1939.fmi`
⌈The value of the attribute `DiagnosticTroubleCodeJ1939.fmi` shall be in the interval 0..31.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1457] Service-only DTCs shall refer to a common memory section ⌈All `DiagnosticTroubleCodeJ1939` with attribute `kind` set to the value `serviceOnly` that reference the same `DiagnosticJ1939Node` shall also reference the same `DiagnosticTroubleCodeProps`.`diagnosticMemory`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1458] Reference to `DiagnosticMemoryDestination` ⌈A `DiagnosticMemoryDestination` that is referenced by a `DiagnosticTroubleCodeJ1939`.`dtcProps`.`diagnosticMemory` where the value of attribute `DiagnosticTroubleCodeJ1939`.`kind` is set to `serviceOnly` shall **not be referenced by any other** `DiagnosticTroubleCodeJ1939` where attribute `kind` is set to any other value than `serviceOnly`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1459] Existence of attributes of `DiagnosticTroubleCodeProps` ⌈The following list of attributes of meta-class `DiagnosticTroubleCodeProps` are not required and therefore shall be ignored if the `DiagnosticTroubleCodeProps` is referenced in the role `dtcProps` from a `DiagnosticTroubleCodeObd`:

- `freezeFrame`
- `snapshotRecordContent`
- `diagnosticMemory`
- `extendedDataRecord`
- `aging`

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1460] Restrictions for the value of `DiagnosticInfoType`.`id` ⌈The values 0x00, 0x20, 0x40, 0x60, 0x80, 0xA0, 0xC0, and 0xE0 are not allowed to appear in the value of `DiagnosticInfoType`.`id`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1461] Restrictions for the value of `DiagnosticTestRoutineIdentifier`.`id` ⌈The values 0x00, 0x20, 0x40, 0x60, 0x80, 0xA0, 0xC0, and 0xE0 are not allowed to appear in the value of `DiagnosticTestRoutineIdentifier`.`id`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1462] Restrictions for the value of `DiagnosticTestResult.testIdentifier.id` ⌈The values 0x00, 0x20, 0x40, 0x60, 0x80, 0xA0, 0xC0, and 0xE0 are not allowed to appear in the value of `DiagnosticTestResult.testIdentifier.id`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1464] Allowed value range of `DiagnosticEnvConditionFormula.nrcValue` ⌈The value of attribute `DiagnosticEnvConditionFormula.nrcValue` shall be limited to the interval [1..255].

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1466] Allowed values of `compareType` in the context of a `DiagnosticEnvModeCondition` ⌈Within the context of a `DiagnosticEnvDataCondition` **only a subset** of the values of `DiagnosticCompareTypeEnum` is supported for the inherited attribute `compareType`, namely:

- `DiagnosticCompareTypeEnum.isEqual`
- `DiagnosticCompareTypeEnum.isNotEqual`

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1467] References in `DiagnosticEnvModeCondition` ⌈In a `DiagnosticEnvModeCondition` the reference `modeElement` shall only point to a `DiagnosticEnvModeElement` that is aggregated inside the same `DiagnosticEnvironmentalCondition` as the `DiagnosticEnvModeCondition` itself.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1470] Value of `DiagnosticAbstractParameter.bitOffset` ⌈The value of `DiagnosticAbstractParameter.bitOffset` shall only be set to a multiple of 8 **at the time when the DEXT is complete**.

⌋*()*

[constr_1472] Existence of `DiagnosticDataIdentifier.supportInfoByte` ⌈The attribute `DiagnosticDataIdentifier.supportInfoByte` **shall not exist** if the value of `DiagnosticDataIdentifier.id` is **outside the range 0xF400-0xF4FF**.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1509] Value of `extendedDataRecord.recordNumber` shall be unique within primary fault memory ⌈For all `DiagnosticTroubleCodeProps` that refer to `DiagnosticMemoryDestinationPrimary` in the role `diagnosticMemory` there shall be no two `extendedDataRecord.recordNumber` with the same value.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1511] Value of `extendedDataRecord.recordNumber` shall be unique within user-defined fault memory ⌈For all `DiagnosticTroubleCodeProps` that refer to `DiagnosticMemoryDestinationUserDefined` in the role `diagnosticMemory` there shall be no two `extendedDataRecord.recordNumber` with the same value for any `DiagnosticMemoryDestinationUserDefined` referenced as `DiagnosticTroubleCodeProps.diagnosticMemory` with a given value of `memoryId`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1512] `freezeFrame.recordNumber` shall be unique within primary fault memory ⌈For all `DiagnosticTroubleCodeProps` that refer to `DiagnosticMemoryDestinationPrimary` in the role `diagnosticMemory` there shall be no two `freezeFrame.recordNumber` with the same value.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1514] `freezeFrame.recordNumber` shall be unique within user-defined fault memory ⌈For all `DiagnosticTroubleCodeProps` that refer to `DiagnosticMemoryDestinationUserDefined` in the role `diagnosticMemory` there shall be no two `freezeFrame.recordNumber` with the same value for any `DiagnosticMemoryDestinationUserDefined` referenced as `DiagnosticTroubleCodeProps.diagnosticMemory` with a given value of `memoryId`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1552] `DiagnosticDataIdentifier` referenced by `DiagnosticDataIdentifierSet` ⌈If a `DiagnosticDataIdentifier` is referenced by `DiagnosticDataIdentifierSet` then the `DiagnosticDataIdentifier` shall not have gaps in between individual elements (as indicated by `DiagnosticParameter.bitOffset` and the length of the aggregated `DiagnosticDataElement`) or at the end of the `DiagnosticDataIdentifier` (as indicated by attribute `DiagnosticDataIdentifier.didSize`).

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1584] `DiagnosticDataElement` **shall not be used more than once in I/O Control instance** ⌈A given `DiagnosticDataElement` shall not be used by more than one `DiagnosticServiceDataMapping` that in turn refers to a `DataPrototype` defined in the context of a `DataInterface` that is used to type a `PortPrototype` that in turn is referenced by a `RoleBasedPortAssignment` where attribute `role` is set to the value `IOControlRequest`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1590] `DiagnosticEvent` **referenced in the role** `masterEvent` **or** `slaveEvent` ⌈Any given `DiagnosticEvent` shall at most once be referenced from a `DiagnosticMasterToSlaveEventMapping`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1591] `DiagnosticEvent` **referenced as** `slaveEvent` **shall not be reported by diagnostic monitor** ⌈A `DiagnosticEvent` referenced in the role `DiagnosticMasterToSlaveEventMapping`.`slaveEvent` shall not be referenced in the role `DiagnosticEventPortMapping`.`diagnosticEvent` and vice versa.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1612] Reference from `DiagnosticRoutineControl` **to** `DiagnosticAccessPermission` **has no meaning** ⌈The reference from `DiagnosticRoutineControl` (via its abstract base class `DiagnosticServiceInstance`) in the role `accessPermission` to meta-class `DiagnosticAccessPermission` shall not be used.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1616] Existence of attribute `DiagnosticExtendedDataRecord.customTrigger` ⌈The attribute `DiagnosticExtendedDataRecord`.`customTrigger` shall only exist if the attribute `DiagnosticExtendedDataRecord`.`trigger` is set to the value `DiagnosticRecordTriggerEnum`.`custom`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1617] Existence of attribute `DiagnosticFreezeFrame.customTrigger` ⌈The attribute `DiagnosticFreezeFrame`.`customTrigger` shall only exist if

the attribute `DiagnosticFreezeFrame.trigger` is set to the value `DiagnosticRecordTriggerEnum.custom`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1623] Restriction on `DiagnosticReadScalingDataByIdentifier.dataIdentifier` ⌈The reference `DiagnosticReadScalingDataByIdentifier.dataIdentifier` shall only refer to a `DiagnosticDataIdentifier`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1624] Existence of `DiagnosticDataElement.scalingInfoSize` ⌈The attribute `DiagnosticDataElement.scalingInfoSize` shall only exist if the enclosing `DiagnosticAbstractParameter` is aggregated by a `DiagnosticDataIdentifier` that is referenced by a `DiagnosticReadScalingDataByIdentifier` in the role `DiagnosticReadScalingDataByIdentifier.dataIdentifier`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1721] `DiagnosticControlEnableMaskBit.bitNumber` shall be unique ⌈Within the context of the enclosing `DiagnosticIOControl` the value of attribute `DiagnosticIOControl.controlEnableMaskBit.bitNumber` shall be unique.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1722] Relation between reference `DiagnosticIOControl.dataIdentifier` and attribute `DiagnosticIOControl.controlEnableMaskBit` ⌈Any `DiagnosticDataElement` referenced in the role `DiagnosticIOControl.controlEnableMaskBit.controlledDataElement` shall be defined in the scope of the `DiagnosticDataIdentifier` that is referenced in the role `DiagnosticIOControl.dataIdentifier`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1745] Indirect reference to `DiagnosticCommonElement` ⌈If a `DiagnosticCommonElement` is referenced from within the context of another `DiagnosticCommonElement` and the referencing `DiagnosticCommonElement` is in turn referenced by a `DiagnosticContributionSet` in the role `element` then the referenced `DiagnosticCommonElement` shall also be referenced by the same `DiagnosticContributionSet`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1749] Existence of `DiagnosticInfoType.dataElement` ⌈For each `DiagnosticInfoType`, at least one aggregation of `DiagnosticParameter` in the role `dataElement` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1750] Existence of attribute `DiagnosticParameterIdentifier.pidSize` ⌈Attribute `DiagnosticParameterIdentifier.pidSize` is only relevant if a gap exists at the end of the `DiagnosticParameterIdentifier`. If this gap does not exist, the size of the `DiagnosticParameterIdentifier` can be computed.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1752] Existence of references owned by `DiagnosticEnableConditionPortMapping` ⌈For each `DiagnosticEnableConditionPortMapping`, only one of the following references

- to `SwcServiceDependency` in the role `swcFlatServiceDependency`
- to `SwcServiceDependency` in the role `swcServiceDependencyInSystem`

may exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1753] Existence of references owned by `DiagnosticStorageConditionPortMapping` ⌈For each `DiagnosticStorageConditionPortMapping`, only one of the following references

- to `SwcServiceDependency` in the role `swcFlatServiceDependency`
- to `SwcServiceDependency` in the role `swcServiceDependencyInSystem`

may exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1756] Existence of attributes `DiagnosticExtendedDataRecord.trigger` and `update` ⌈For each `DiagnosticExtendedDataRecord`, attributes `trigger` and `update` shall only exist **at the time when the DEXT is complete** if at least one `DiagnosticDataElement` is aggregated by a `DiagnosticExtendedDataRecord.recordElement` in the role `dataElement` to which no reference in the role `DiagnosticDemProvidedDataMapping.dataElement` exists.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1757] Existence of attribute `DiagnosticTroubleCodeUds.udsDtcValue` ⌈For each `DiagnosticTroubleCodeUds`, attribute `udsDtcValue` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1758] Existence of attribute `DiagnosticTroubleCodeObd.obdDTCValue` ⌈For each `DiagnosticTroubleCodeObd`, attribute `obdDTCValue` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1759] Existence of references owned by `DiagnosticOperationCyclePortMapping` ⌈For each `DiagnosticOperationCyclePortMapping`, only one of the following references

- to `SwcServiceDependency` in the role `swcFlatServiceDependency`

- to `SwcServiceDependency` in the role `swcServiceDependencyInSystem`

shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1760] Existence of `DiagnosticExtendedDataRecord.recordElement` ⌈For each `DiagnosticExtendedDataRecord`, at least one aggregation of `DiagnosticParameter` in the role `recordElement` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1761] Existence of attribute `DiagnosticConnectedIndicator.healingCycle` ⌈`DiagnosticConnectedIndicator.healingCycle` shall **only exist** if the value of `DiagnosticConnectedIndicator.healingCycleCounterThreshold` is **not equal to 0**.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1762] Existence of references owned by `DiagnosticEventPortMapping` ⌈For each `DiagnosticEventPortMapping`, only one of the references

- to `BswServiceDependency` in the role `bswServiceDependency`

- to `SwcServiceDependency` in the role `swcFlatServiceDependency`

- to `SwcServiceDependency` in the role `swcServiceDependencyInSystem`

shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1763] **Existence of attribute `DiagnosticPeriodicRate.periodicRateCategory`** ⌈For each `DiagnosticPeriodicRate`, the attribute `periodicRateCategory` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1766] Existence of `DiagEventDebounceCounterBased.counterJumpDownValue` ⌈For each `DiagEventDebounceCounterBased`, attribute `counterJumpDownValue` shall only exist **at the time when the DEXT is complete** if attribute `counterJumpDown` exists and is set to `True`.

⌋*()*

[constr_1767] **Existence of `DiagEventDebounceCounterBased.counterJumpUpValue`** ⌈For each `DiagEventDebounceCounterBased`, attribute `counterJumpUpValue` shall only exist **at the time when the DEXT is complete** if attribute `counterJumpUp` exists and is set to `True`.

⌋*()*

[constr_1768] Existence of attribute `DiagnosticEvent.associatedEventIdentification` ⌈Attribute `DiagnosticEvent.associatedEventIdentification` shall exist if the respective `DiagnosticEvent` is mapped to a `DiagnosticTroubleCodeUds` and one of the following conditions is fulfilled:

- The reference `DiagnosticTroubleCodeUds.dtcProps.snapshotRecordContent` exists and the referenced `DiagnosticDataIdentifierSet` references at least one `dataIdentifier.dataElement.dataElement` that is also referenced by a `DiagnosticDemProvidedDataMapping` that has attribute `dataProvider` set to the value `DEM_EVENT_ASSOCIATED_IDENTIFICATION`.

- The reference `DiagnosticTroubleCodeUds.dtcProps.extendedDataRecord` exists and the referenced `DiagnosticExtendedDataRecord` aggregates at least one `recordElement.dataElement` that is also referenced by a `DiagnosticDemProvidedDataMapping` that has attribute `dataProvider` set to the value `DEM_EVENT_ASSOCIATED_IDENTIFICATION`.

  This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1772] Unique `DiagnosticSession` and `DiagnosticSecurityLevel` for diagnostic routines that have the same identifier ⌈All `DiagnosticAccessPermission`s referenced from `DiagnosticRoutine`s where attribute `DiagnosticRoutine.id` has the identical value shall refer to the identical set of `DiagnosticSession` and `DiagnosticSecurityLevel`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1780] Existence of attribute `DiagnosticTroubleCodeJ1939.fmi` ⌈For each `DiagnosticTroubleCodeJ1939`, attribute `fmi` shall exist **at the time when the DEXT is complete.**

⌋*()*

[constr_1781] Existence of attribute `DiagnosticTroubleCodeJ1939.spn` ⌈For each `DiagnosticTroubleCodeJ1939`, attribute `spn` shall exist **at the time when the DEXT is complete.**

⌋*()*

[constr_1782] Usage of internal data elements only for extended data records ⌈A `DiagnosticDemProvidedDataMapping` shall **only** refer to a `DiagnosticDataElement` that is aggregated by a `DiagnosticExtendedDataRecord` in the role `recordElement.dataElement`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_1790] Existence of attribute `DiagnosticAbstractParameter.bitOffset` ⌈For each `DiagnosticParameter`, attribute `bitOffset` shall exist **at the time when the DEXT is complete.**

⌋*()*

[constr_1791] Existence of attribute `dataElement` vs. `parameterSize` of metaclass `DiagnosticParameter` ⌈For each `DiagnosticParameter`, exactly **one** of the attributes

- `dataElement` or
- `parameterSize`

shall exist **at the time when the DEXT is complete.**

⌋*()*

[constr_1792] Existence of `DiagnosticDataIdentifier.dataElement` ⌈For each `DiagnosticDataIdentifier`, the aggregation of `DiagnosticParameter` in the role `dataElement` shall exist at lest once **at the time when the DEXT is complete.**

⌋*()*

[constr_1793] Existence of attribute `DiagnosticAbstractDataIdentifier.id` ⌈For each `DiagnosticAbstractDataIdentifier`, attribute `id` shall exist **at the time when the DEXT is complete.**

⌋*()*

[constr_1794] Existence of attribute `DiagnosticProtocol.priority` ⌈For each `DiagnosticProtocol`, attribute `priority` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1795] Existence of attribute `DiagnosticProtocol.protocolKind` ⌈For each `DiagnosticProtocol`, attribute `protocolKind` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1796] Existence of attribute `DiagnosticServiceTable.serviceInstance` ⌈For each `DiagnosticServiceTable`, attribute `serviceInstance` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1797] Existence of attribute `DiagnosticServiceTable.protocolKind` ⌈For each `DiagnosticServiceTable`, attribute `protocolKind` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1798] Existence of `DiagnosticServiceInstance.serviceClass` ⌈For each subclass of `DiagnosticServiceInstance`, a reference with the abstract role `serviceClass` shall exist **at the time when the DEXT is complete** to a matching subclass of `DiagnosticServiceClass`.

This rule applies unless a rule for a specific combination of matching sub-classes of `DiagnosticServiceInstance` and `DiagnosticServiceClass` exists.

⌋*()*

[constr_1799] Existence of `DiagnosticEnvironmentalCondition.formula` ⌈For each `DiagnosticEnvironmentalCondition`, the aggregation of `DiagnosticEnvConditionFormula` in the role `formula` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1800] Existence of `DiagnosticEnvConditionFormula.op` ⌈For each `DiagnosticEnvConditionFormula`, that attribute `op` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1801] Existence of `DiagnosticEnvCompareCondition.compareType` ⌈For each `DiagnosticEnvCompareCondition`, that attribute `compareType` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1802] Existence of `DiagnosticEnvDataCondition.compareValue` ⌈For each `DiagnosticEnvDataCondition`, that attribute `compareValue` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1803] Existence of `DiagnosticEnvDataCondition.dataElement` ⌈For each `DiagnosticEnvDataCondition`, that attribute `dataElement` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1804] Existence of `DiagnosticEnvModeCondition.modeElement` ⌈For each `DiagnosticEnvModeCondition`, that attribute `modeElement` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1805] Existence of `DiagnosticEnvSwcModeElement.mode` ⌈For each `DiagnosticEnvSwcModeElement`, that attribute `mode` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1806] Existence of `DiagnosticEnvBswModeElement.mode` ⌈For each `DiagnosticEnvBswModeElement`, that attribute `mode` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1807] Existence of reference `DiagnosticDataByIdentifier.dataIdentifier` ⌈For each `DiagnosticDataByIdentifier`, the reference `dataIdentifier` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1808] Existence of reference `DiagnosticDynamicallyDefineDataIdentifier.dataIdentifier` ⌈For each `DiagnosticDynamicallyDefineDataIdentifier`, the reference to `DiagnosticDynamicDataIdentifier` in the role `dataIdentifier` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1810] Existence of aggregation `DiagnosticReadDataByPeriodicIDClass.periodicRate` ⌈For each `DiagnosticReadDataByPeriodicIDClass`, the aggregation of `DiagnosticPeriodicRate` in the role `periodicRate` shall exist at least once **at the time when the DEXT is complete**.

⌋*()*

[constr_1811] Existence of attribute `DiagnosticReadDataByPeriodicIDClass.maxPeriodicDidToRead` ⌈For each `DiagnosticReadDataByPeriodi-`

cIDClass, the attribute `maxPeriodicDidToRead` shall exist at least once **at the time when the DEXT is complete**.

⌋*()*

[constr_1812] Existence of attribute `DiagnosticReadDataByPeriodicIDClass.schedulerMaxNumber` ⌈For each `DiagnosticReadDataByPeriodicIDClass`, the attribute `schedulerMaxNumber` shall exist at least once **at the time when the DEXT is complete**.

⌋*()*

[constr_1815] Existence of attribute `DiagnosticRoutine.id` ⌈For each `DiagnosticRoutine`, the attribute `id` shall exist at least once **at the time when the DEXT is complete**.

⌋*()*

[constr_1816] Existence of attribute `DiagnosticSecurityAccess.requestSeedId` ⌈For each `DiagnosticSecurityAccess`, the attribute `requestSeedId` shall exist at least once **at the time when the DEXT is complete**.

⌋*()*

[constr_1817] Existence of attribute `DiagnosticSecurityAccess.securityLevel` ⌈For each `DiagnosticSecurityAccess`, the attribute `securityLevel` shall exist at least once **at the time when the DEXT is complete**.

⌋*()*

[constr_1818] Existence of reference `DiagnosticSessionControl.diagnosticSession` ⌈For each `DiagnosticSessionControl`, the reference to `DiagnosticSession` in the role `diagnosticSession` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1819] Existence of attribute `DiagnosticParameterIdentifier.id` ⌈For each `DiagnosticParameterIdentifier`, attribute `id` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1820] Existence of reference `DiagnosticRequestCurrentPowertrainData.pid` ⌈For each `DiagnosticRequestCurrentPowertrainData`, the reference to `DiagnosticParameterIdentifier` in the role `pid` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1821] Existence of reference `DiagnosticRequestPowertrainFreezeFrameData.freezeFrame` ⌈For each `DiagnosticRequestPowertrainFreeze-

`FrameData`, the reference to `DiagnosticParameterIdentifier` in the role `freezeFrame` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1822] Existence of reference `DiagnosticRequestControlOfOnBoard-Device.testId` ⌈For each `DiagnosticRequestControlOfOnBoardDevice`, the reference to `DiagnosticParameterIdentifier` in the role `testId` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1823] Existence of attribute `DiagnosticTestRoutineIdentifier.id` ⌈For each `DiagnosticTestRoutineIdentifier`, attribute `id` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1824] Existence of attribute `DiagnosticTestRoutineIdentifier.requestDataSize` ⌈For each `DiagnosticTestRoutineIdentifier`, attribute `requestDataSize` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1825] Existence of attribute `DiagnosticTestRoutineIdentifier.responseDataSize` ⌈For each `DiagnosticTestRoutineIdentifier`, attribute `responseDataSize` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1826] Existence of reference `DiagnosticRequestVehicleInfo.infoType` ⌈For each `DiagnosticRequestVehicleInfo`, the reference to `DiagnosticParameterIdentifier` in the role `infoType` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1827] Existence of attribute `DiagnosticInfoType.id` ⌈For each `DiagnosticInfoType`, attribute `id` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1828] Existence of referenced from `DiagnosticServiceDataMapping` ⌈For each `DiagnosticServiceDataMapping`, the following references shall exist **at the time when the DEXT is complete**:

- Reference to a `DiagnosticServiceMappingDiagTarget`, i.e. one of

    - Reference to `DiagnosticDataElement` in the role `diagnosticDataElement`

- Reference to `DiagnosticParameterIdent` in the role `diagnosticParameter`

- Reference to `DiagnosticParameterElement` in the roles

  * `contextElement` (optional)

  * `targetElement`

  from within `parameterElementAccess`

- Reference to `DataPrototype` in the role `mappedDataElement`

⌋*()*

[constr_1829] Existence of reference `DiagnosticConnectedIndicator.indicator` ⌈For each `DiagnosticConnectedIndicator`, the reference to `DiagnosticIndicator` in the role `indicator` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1830] Existence of `DiagnosticTroubleCodeGroup.groupNumber` ⌈For each `DiagnosticTroubleCodeGroup`, attribute `groupNumber` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1831] Existence of `DiagnosticTroubleCodeProps.priority` ⌈For each `DiagnosticTroubleCodeProps`, attribute `priority` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1832] Existence of `DiagnosticExtendedDataRecord.recordNumber` ⌈For each `DiagnosticExtendedDataRecord`, attribute `recordNumber` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1833] Existence of `DiagnosticFreezeFrame.trigger` ⌈For each `DiagnosticFreezeFrame`, attribute `trigger` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1834] Existence of `DiagnosticCondition.initValue` ⌈For each `DiagnosticCondition`, attribute `initValue` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1835] Existence of `DiagEventDebounceCounterBased.counterDecrementStepSize` ⌈For each `DiagEventDebounceCounterBased`, attribute

counterDecrementStepSize shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1836] Existence of `DiagEventDebounceCounterBased.counterIncrementStepSize` ⌈For each `DiagEventDebounceCounterBased`, attribute `counterIncrementStepSize` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1837] Existence of `DiagEventDebounceCounterBased.counterFailedThreshold` ⌈For each `DiagEventDebounceCounterBased`, attribute `counterFailedThreshold` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1838] Existence of `DiagEventDebounceCounterBased.counterPassedThreshold` ⌈For each `DiagEventDebounceCounterBased`, attribute `counterPassedThreshold` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1839] Existence of attribute `DiagEventDebounceTimeBased.timeFailedThreshold` ⌈For each `DiagEventDebounceTimeBased`, attribute `timeFailedThreshold` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1840] Existence of attribute `DiagEventDebounceTimeBased.timePassedThreshold` ⌈For each `DiagEventDebounceTimeBased`, attribute `timePassedThreshold` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1841] Existence of `DiagnosticEnableConditionGroup.enableCondition` ⌈For each `DiagnosticEnableConditionGroup`, attribute `enableCondition` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1842] Existence of `DiagnosticStorageConditionGroup.storageCondition` ⌈For each `DiagnosticStorageConditionGroup`, attribute `storageCondition` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1843] Existence of reference `DiagnosticEventPortMapping.diagnosticEvent` ⌈For each `DiagnosticEventPortMapping`, the reference to `Diag-`

nosticEvent in the role diagnosticEvent shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1844] Existence of reference `DiagnosticOperationCyclePortMapping.operationCycle` ⌈For each DiagnosticOperationCyclePortMapping, the reference to DiagnosticOperationCycle in the role operationCycle shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1845] Existence of reference `DiagnosticEnableConditionPortMapping.enableCondition` ⌈For each DiagnosticEnableConditionPortMapping, the reference to DiagnosticEnableCondition in the role enableCondition shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1846] Existence of reference `DiagnosticStorageConditionPortMapping.diagnosticStorageCondition` ⌈For each DiagnosticStorageConditionPortMapping, the reference to DiagnosticStorageCondition in the role diagnosticStorageCondition shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1847] Existence of reference `DiagnosticDemProvidedDataMapping.dataElement` ⌈For each DiagnosticDemProvidedDataMapping, the reference to DiagnosticDataElement in the role dataElement shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1848] Existence of attribute `DiagnosticAging.agingCycle` ⌈For each DiagnosticAging, attribute agingCycle shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1849] Existence of attribute `DiagnosticAging.threshold` ⌈For each DiagnosticAging, attribute threshold shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1850] Existence of aggregation `DiagnosticTestResult.testIdentifier` ⌈For each DiagnosticTestResult, the aggregation of meta-class DiagnosticTestIdentifier in the role testIdentifier shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1851] Existence of reference `DiagnosticTestResult.monitoredIdentifier` ⌈For each `DiagnosticTestResult`, the reference to meta-class `DiagnosticTestIdentifier` in the role `monitoredIdentifier` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1852] Existence of attribute `DiagnosticEcuInstanceProps.obdSupport` ⌈For each `DiagnosticEcuInstanceProps`, attribute `obdSupport` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1853] Existence of attribute `DiagnosticIumprGroup.iumprGroupIdentifier` ⌈For each `DiagnosticIumprGroup`, attribute `iumprGroupIdentifier` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1854] Existence of attribute `DiagnosticIumprGroupIdentifier.groupId` ⌈For each `DiagnosticIumprGroupIdentifier`, attribute `groupId` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1855] Existence of attribute `DiagnosticFunctionIdentifierInhibit.inhibitionMask` ⌈For each `DiagnosticFunctionIdentifierInhibit`, attribute `inhibitionMask` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1856] Existence of attribute `DiagnosticJ1939Spn.spn` ⌈For each `DiagnosticJ1939Spn`, attribute `spn` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1857] Existence of the reference `DiagnosticEventToTroubleCodeJ1939Mapping.diagnosticEvent` ⌈For each `DiagnosticEventToTroubleCodeJ1939Mapping`, reference `diagnosticEvent` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1858] Existence of the attribute `DiagnosticEventToTroubleCodeJ1939Mapping.troubleCodeJ1939` ⌈For each `DiagnosticEventToTroubleCodeJ1939Mapping`, attribute `troubleCodeJ1939` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_1859] Usage of `DiagnosticRecordTriggerEnum.testFailedThisOperationCycle` ⌈The enumeration value `DiagnosticRecordTriggerEnum.testFailedThisOperationCycle` shall only be used in the context of meta-class `DiagnosticFreezeFrame`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_10024]{DRAFT} **Existence of reference in the role `DiagnosticSecurityEventReportingModeMapping.dataElement`** ⌈For each `DiagnosticSecurityEventReportingModeMapping`, the reference to `DiagnosticDataElement` in the role `dataElement` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_10025]{DRAFT} **Existence of reference in the role `DiagnosticSecurityEventReportingModeMapping.securityEvent`** ⌈For each `DiagnosticSecurityEventReportingModeMapping`, the reference to `SecurityEventContextProps` in the role `securityEvent` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_10026]{DRAFT} **Existence of reference in the role `DiagnosticEventToSecurityEventMapping.diagnosticEvent`** ⌈For each `DiagnosticEventToSecurityEventMapping`, the reference to `DiagnosticEvent` in the role `diagnosticEvent` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_10027]{DRAFT} **Existence of reference in the role `DiagnosticEventToSecurityEventMapping.securityEventProps`** ⌈For each `DiagnosticEventToSecurityEventMapping`, the reference to `SecurityEventContextProps` in the role `securityEventProps` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_10038] Restriction for the usage of `DiagnosticAccessPermission.authenticationRole` ⌈Attribute `DiagnosticAccessPermission.authenticationRole` shall not exist if the `DiagnosticAccessPermission` is referenced from

- `DiagnosticRequestCurrentPowertrainData`

- `DiagnosticRequestPowertrainFreezeFrameData`

- `DiagnosticRequestEmissionRelatedDTC`

- `DiagnosticClearResetEmissionRelatedInfo`

- `DiagnosticRequestOnBoardMonitoringTestResults`

- DiagnosticRequestControlOfOnBoardDevice

- DiagnosticRequestVehicleInfo

- DiagnosticRequestEmissionRelatedDTCPermanentStatus

- sub-classes of DiagnosticAuthentication

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_10042] Existence of attribute DiagnosticCommonProps.defaultEndianness ⌈One of the following conditions shall be fulfilled **at the time when the DEXT is complete**:

- DiagnosticCommonProps.defaultEndianness exists.

- The attribute DiagnosticParameter.dataElement.swDataDefProps. baseType.baseTypeDefinition.baseTypeEncoding exist for **all** DiagnosticParameters defined in the context of the DiagnosticContributionSet.

⌋*()*

[constr_10043] Existence of attribute DiagnosticCommonProps.resetConfirmedBitOnOverflow ⌈Attribute DiagnosticCommonProps.resetConfirmedBitOnOverflow shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_10044] Existence of attribute DiagnosticCommonProps.occurrenceCounterProcessing ⌈If, in the context of a DiagnosticContributionSet, a DiagnosticDemProvidedDataMapping exists where attribute DiagnosticDemProvidedDataMapping.dataProvider is set to the value DEM_OCCCTR, then attribute DiagnosticCommonProps.occurrenceCounterProcessing shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_10045] Existence of attribute DiagnosticSecurityAccessClass.securityDelayTimeOnBoot ⌈Attribute DiagnosticSecurityAccessClass.securityDelayTimeOnBoot shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_10084] Existence of DiagnosticIumprToFunctionIdentifierMapping.iumpr ⌈For all DiagnosticIumprToFunctionIdentifierMapping, the reference in the role iumpr shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_10085] **Existence of `DiagnosticIumprToFunctionIdentifierMapping.functionIdentifier`** ⌈For all `DiagnosticIumprToFunctionIdentifierMapping`, the reference in the role `functionIdentifier` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_10088] Relation between event and DTC without event combination ⌈If attribute `DiagnosticCommonProps.typeOfEventCombinationSupported` is not configured, then all `DiagnosticTroubleCodeUds` that refer to a `DiagnosticTroubleCodeProps` in the role `dtcProps` that in turn refers to a `DiagnosticMemoryDestination` in the role `diagnosticMemory` shall only be referenced by at most one `DiagnosticEventToTroubleCodeUdsMapping`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_10089] **Existence of attribute `DiagnosticCommonProps.eventCombinationReportingBehavior`** ⌈Attribute `DiagnosticCommonProps.eventCombinationReportingBehavior` is always optional and shall be set to the value `DiagnosticEventCombinationReportingBehaviorEnum.reportingInChronlogicalOrderOldestFirst` only if attribute `DiagnosticCommonProps.typeOfEventCombinationSupported` is set to the value `DiagnosticEventCombinationBehaviorEnum.eventCombinationOnRetrieval`.

If it is missing, then the reporting order is not specified. This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_10091] Mandatory subfunction of diagnostic service `Authentication` ⌈If the diagnostic service `Authentication` is supported, then the following subfunctions shall be configured:

- De-authentication, formalized by meta-class `DiagnosticDeAuthentication`.

- Proof of ownership, formalized by meta-class `DiagnosticProofOfOwnership`.

- Authentication configuration, formalized by meta-class `DiagnosticAuthenticationConfiguration`.

- One of

  - Verify certificate unidirectional, formalized by meta-class `DiagnosticVerifyCertificateUnidirectional`.

  - Verify certificate bidirectional, formalized by meta-class `DiagnosticVerifyCertificateBidirectional`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_10100] Existence of `DiagnosticRoutineControl.routine` ⌈For each `DiagnosticRoutineControl`, the attribute `routine` shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_10115] Existence of attributes of `DiagnosticEnvDataElementCondition` if the reference in the role `dataPrototype` exists ⌈If the reference in the role `DiagnosticEnvDataElementCondition.dataPrototype` exists, then

- the aggregation in the role `compareValue` shall exist and
- the aggregation in the role `swDataDefProps` shall not exist

at the time when the DEXT is complete.

⌋*()*

[constr_10116] Existence of attributes of `DiagnosticEnvDataElementCondition` if the reference in the role `dataPrototype` does not exist ⌈If the reference in the role `DiagnosticEnvDataElementCondition.dataPrototype` does **not** exist, then the aggregations in the role

- `compareValue` and
- `swDataDefProps`

shall exist **at the time when the DEXT is complete**.

⌋*()*

[constr_10117] Existence of attributes of `DiagnosticEnvDataElementCondition.swDataDefProps` ⌈

Attribute of `SwDataDefProps`	Attribute Existence
`baseType`	1
`compuMethod`	0..1
`dataConstr`	0..1

⌋*()*

[constr_10122] Existence of attribute `DiagnosticComControlSubNodeChannel.subNodeChannel` ⌈Attribute `DiagnosticComControlSubNodeChannel.subNodeChannel` shall only exist if the value of `DiagnosticComControl.category` is set to either

- `ENABLE_RX_AND_DISABLE_TX_WITH_ENHANCED_ADDRESS_INFORMATION` or
- `ENABLE_RX_AND_TX_WITH_ENHANCED_ADDRESS_INFORMATION`.

This rule shall be imposed **at the time when the DEXT is complete**

⌋*()*

[constr_10364] Usage of DiagnosticRecordTriggerEnum.testPassed ⌈The enumeration value DiagnosticRecordTriggerEnum.testPassed shall only be used in context of DiagnosticExtendedDataRecord.trigger **at the time when the DEXT is complete**

⌋*()*

[constr_10368] Restriction regarding the reference DiagnosticDataIdentifierSet.dataIdentifier ⌈A DiagnosticDataIdentifier that is referenced in the role DiagnosticDataIdentifierSet.dataIdentifier shall not aggregate in the role dataElement a DiagnosticParameter that aggregates in the role ident a DiagnosticParameterIdent that in turn aggregates in the role subElement a DiagnosticParameterElement.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_10369] Existence of attributes of DiagnosticParameterElement depending on the value of attribute category ⌈

Value of category	Description	array-Size	subElement	dataElement
LEAF	The DiagnosticParameterElement represents a "leaf" element of a nested structure.	No	No	Yes
ARRAY	The DiagnosticParameterElement represents an array, i.e references to this DiagnosticParameterElement shall define a value for the attribute index. Arrays of "primitive" types are defined in the context of the DiagnosticDataElement	Yes	Yes, if dataElement does not exist	Yes, if subElement does not exist
STRUCTURE	The DiagnosticParameterElement represents a structure with one or more elements.	No	Yes	No

⌋*()*

[constr_10370] Restriction regarding the role DiagnosticParameterIdentifier.dataElement ⌈A DiagnosticParameter that is aggregated by a DiagnosticParameterIdentifier in the role dataElement shall not aggregate in the role ident a DiagnosticParameterIdent that in turn aggregates in the role subElement a DiagnosticParameterElement.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

[constr_10371] Restriction regarding the aggregation of DiagnosticParameter in the role DiagnosticExtendedDataRecord.recordElement ⌈A DiagnosticParameter that is aggregated by a DiagnosticExtendedDataRecord in the role recordElement shall not aggregate in the role ident a DiagnosticParameterI-

dent that in turn aggregates in the role `subElement` a `DiagnosticParameterElement`.

This rule shall be imposed **at the time when the DEXT is complete**.

⌋*()*

2.3 TPS_ECUConfiguration

[constr_3022] `EcucModuleDef` **category restriction** ⌈The category definition shall be restricted to exactly the two defined ones:

- VENDOR_SPECIFIC_MODULE_DEFINITION
- STANDARDIZED_MODULE_DEFINITION

⌋*()*

[constr_3023] Usage of `apiServicePrefix` ⌈The attribute `apiServicePrefix` is mandatory for VSMDs derived from the CDD and Xfrm StMD. The attribute shall not be provided for VSMDs derived from any other StMDs.

⌋*()*

[constr_3091] Multiplicity of `valueConfigClass` ⌈The multiplicity of the attribute `EcucCommonAttributes.valueConfigClass` shall not exceed 3.

⌋*()*

[constr_3092] Usage of `configVariant` **and** `configClass` **attributes** ⌈`configVariant` and `configClass` shall always exist as a pair for each existing `EcucAbstractConfigurationClass` (`EcucValueConfigurationClass` or `EcucMultiplicityConfigurationClass` depending on the context).

⌋*()*

[constr_3119] Necessary content of `EcucDestinationUriDef`**s that are referenced by an** `EcucContainerDef` ⌈The `EcucDestinationUriDef` that is referenced by the `EcucContainerDef` in the role `destinationUri` shall define at least the analogous set of `container`s, `parameter`s and `reference`s defined by the `EcucDestinationUriPolicy` of the `EcucDestinationUriDef` that is referenced by the `EcucUriReferenceDef` that targets the `EcucContainerDef`.

⌋*()*

[constr_3120] Applicable attributes when `destinationUriNestingContract` **is set to** `targetContainer` ⌈If the `destinationUriNestingContract` is set to `targetContainer` the attributes `parameter` and `reference` shall not exist.

⌋*()*

[constr_3200] Restriction on values of `EcucDefinitionElement.related-TraceItem` in the VSMD ⌈The value of `EcucDefinitionElement.relatedTraceItem` in the VSMD shall never start with 'ECUC_'.

⌋*()*

[constr_3217] Symbolic name reference shall point only to containers with a symbolic name value defined ⌈If an `EcucReferenceValue` exists that refers in the role `definition` to an `EcucAbstractInternalReferenceDef` with the attribute `requiresSymbolicNameValue` set to true, then the `EcucContainerValue` that is the target of the reference shall refer to an `EcucParamConfContainerDef` in the role `definition` that contains a definition of an `EcucParameterDef` where the attribute `symbolicNameValue` exists and is set to true. The `EcucContainerValue` shall define an `EcucParameterValue` that refers to an `EcucParameterDef` where the attribute `symbolicNameValue` exists and is set to true.

⌋*()*

[constr_3233] `EcucModuleDef` that relies on `EcucCommonAttributes` with `valueConfigClass` set to `Link`/`PostBuild` of another `EcucModuleDef` ⌈If one `EcucModuleDef` relies on the `EcucCommonAttributes` (parameters and references) with `valueConfigClass.configClass` set to `Link`/`PostBuild` of another `EcucModuleDef`, the values of these `EcucCommonAttributes` can only be changed at `Link`/`PostBuild` time if the corresponding `EcucModuleConfigurationValues` of the using `EcucModuleDef` has the `implementationConfigVariant` set to `VariantLinkTime`/`VariantPostBuild`, respectively.

⌋*()*

[constr_3234] `EcucModuleDef` that relies on `EcucCommonAttributes` with `multiplicityConfigClass` set to `Link`/`PostBuild` of another `EcucModuleDef` ⌈If one `EcucModuleDef` relies on the `EcucCommonAttributes` (parameters and references) with `multiplicityConfigClass.configClass` set to `Link`/`PostBuild` of another `EcucModuleDef`, the number of instances of these `EcucCommonAttributes` can only be changed at `Link`/`PostBuild` time if the corresponding `EcucModuleConfigurationValues` of the using `EcucModuleDef` has the `implementationConfigVariant` set to `VariantLinkTime`/`VariantPostBuild`, respectively.

⌋*()*

[constr_3235] `EcucModuleDef` that relies on `EcucContainerDef`s with `multiplicityConfigClass` set to `Link`/`PostBuild` of another `EcucModuleDef` ⌈If one `EcucModuleDef` relies on the `EcucContainerDef`s with `multiplicityConfigClass.configClass` set to `Link`/`PostBuild` of another `EcucModuleDef`, the number of instances of these `EcucContainerDef`s can only be changed at `Link`/`PostBuild` time if the corresponding `EcucModuleConfigurationValues` of the

using `EcucModuleDef` has the `implementationConfigVariant` set to `VariantLinkTime`/`VariantPostBuild`, respectively.

⌋*()*

[constr_3236] `EcucModuleDef` **that relies on** `EcucCommonAttributes` **with** `postBuildVariantValue` **set to true of another** `EcucModuleDef` ⌈If one `EcucModuleDef` relies on the `EcucCommonAttributes` (parameters and references) with `postBuildVariantValue` set to true of another `EcucModuleDef`, the values of these `EcucCommonAttributes` can only differ in different post-build variants if the implementation of the using `EcucModuleDef` supports post-build variations.

⌋*()*

[constr_3237] `EcucModuleDef` **that relies on** `EcucCommonAttributes` **with** `postBuildVariantMultiplicity` **set to true of another** `EcucModuleDef` ⌈If one `EcucModuleDef` relies on the `EcucCommonAttributes` (parameters and references) with `postBuildVariantMultiplicity` set to true of another `EcucModuleDef`, the number of instances of these `EcucCommonAttributes` can only differ in different post-build variants if the implementation of the using `EcucModuleDef` supports post-build variations.

⌋*()*

[constr_3238] `EcucModuleDef` **that relies on** `EcucContainerDef` **with** `postBuildVariantMultiplicity` **set to true of another** `EcucModuleDef` ⌈If one `EcucModuleDef` relies on the `EcucContainerDef`s with `postBuildVariantMultiplicity` set to true of another `EcucModuleDef`, the number of instances of these `EcucContainerDef`s can can only differ in different post-build variants if the implementation of the using `EcucModuleDef` supports post-build variations.

⌋*()*

[constr_3307] ShortNames of `PredefinedVariant`**s referenced by** `EcucPostBuildVariantRef`**s** ⌈All `PredefinedVariant`s that are referenced by `EcucPostBuildVariantRef`s shall have different `shortName`s.

⌋*()*

[constr_3449] Impact of `postBuildVariantUsed` **value set to FALSE** ⌈If the value of the `EcucModuleConfigurationValues.postBuildVariantUsed` is set to FALSE or if it is not defined, it is not possible to add a post-build variant at post-build configuration time.

⌋*()*

[constr_3450] `postBuildVariantUsed` **value in case of post build** `VariationPoint`**s** ⌈If the configuration values of a BSW module contain at least one post build `VariationPoint`, the value of the `postBuildVariantUsed` for the `EcucModuleConfigurationValues` shall be set to TRUE.

⌋*()*

[constr_3451] `EcucModuleConfigurationValues.postBuildVariantUsed` **value setting restriction in case** `postBuildVariantSupport` **is set to TRUE** ⌈If `EcucModuleDef.postBuildVariantSupport` is set to TRUE, then `EcucModuleConfigurationValues.postBuildVariantUsed` can be either TRUE or FALSE. ⌋*()*

[constr_3452] `EcucModuleConfigurationValues.postBuildVariantUsed` **value setting restriction in case** `postBuildVariantSupport` **is set to FALSE** ⌈If `EcucModuleDef.postBuildVariantSupport` is set to FALSE, then `EcucModuleConfigurationValues.postBuildVariantUsed` shall be FALSE. ⌋*()*

[constr_3509] Applicability of `scope` **attribute** ⌈The usage of the attribute `scope` is prohibited for `EcucModuleDef` and for sub-classes of `EcucContainerDef` (i.e. `EcucChoiceContainerDef` and `EcucParamConfContainerDef`). ⌋*()*

[constr_3570] `EcucDefinitionElement.lowerMultiplicity` **always required** ⌈The attribute `EcucDefinitionElement.lowerMultiplicity` shall always be defined **when the ECU Configuration Parameter definition is complete**. ⌋*()*

[constr_3571] `EcucCommonAttributes.origin` **always required** ⌈The attribute `EcucCommonAttributes.origin` shall always be defined **when the ECU Configuration Parameter definition is complete**. ⌋*()*

[constr_3572] `EcucParameterDef.symbolicNameValue` **always required** ⌈The attribute `EcucParameterDef.symbolicNameValue` shall always be defined **when the ECU Configuration Parameter definition is complete**. ⌋*()*

[constr_3573] `EcucAbstractConfigurationClass.configClass` **always required** ⌈The attribute `EcucAbstractConfigurationClass.configClass` shall always be defined **when the ECU Configuration Parameter definition is complete**. ⌋*()*

[constr_3574] `EcucAbstractConfigurationClass.configVariant` **always required** ⌈The attribute `EcucAbstractConfigurationClass.configVariant` shall always be defined **when the ECU Configuration Parameter definition is complete**. ⌋*()*

[constr_3575] `EcucEnumerationLiteralDef.origin` **always required** ⌈The attribute `EcucEnumerationLiteralDef.origin` shall always be defined **when the ECU Configuration Parameter definition is complete**.

⌋*()*

[constr_3576] `EcucInstanceReferenceDef.destinationContext` **always required** ⌈The attribute `EcucInstanceReferenceDef.destinationContext` shall always be defined **when the ECU Configuration Parameter definition is complete**.

⌋*()*

[constr_3577] `EcucInstanceReferenceDef.destinationType` **always required** ⌈The attribute `EcucInstanceReferenceDef.destinationType` shall always be defined **when the ECU Configuration Parameter definition is complete**.

⌋*()*

[constr_3578] `EcucForeignReferenceDef.destinationType` **always required** ⌈The attribute `EcucForeignReferenceDef.destinationType` shall always be defined **when the ECU Configuration Parameter definition is complete**.

⌋*()*

[constr_3579] `EcucReferenceDef.destination` **always required** ⌈The attribute `EcucReferenceDef.destination` shall always be defined **when the ECU Configuration Parameter definition is complete**.

⌋*()*

[constr_3580] `EcucUriReferenceDef.destinationUri` **always required** ⌈The attribute `EcucUriReferenceDef.destinationUri` shall always be defined **when the ECU Configuration Parameter definition is complete**.

⌋*()*

[constr_3581] `EcucDestinationUriDefSet.destinationUriDef` **always required** ⌈The attribute `EcucDestinationUriDefSet.destinationUriDef` shall always be defined **when the ECU Configuration Parameter definition is complete**.

⌋*()*

[constr_3582] `EcucDestinationUriDef.destinationUriPolicy` **always required** ⌈The attribute `EcucDestinationUriDef.destinationUriPolicy` shall always be defined **when the ECU Configuration Parameter definition is complete**.

⌋*()*

[constr_3583] `EcucDestinationUriPolicy.destinationUriNestingContract` **always required** ⌈The attribute `EcucDestinationUriPolicy.destinationUriNestingContract` shall always be defined **when the ECU Configuration Parameter definition is complete**.

⌋*()*

[constr_3584] `EcucQuery`.`ecucQueryExpression` **always required** ⌈The attribute `EcucQuery`.`ecucQueryExpression` shall always be defined **when the ECU Configuration Parameter definition is complete**.

⌋*()*

[constr_3585] `EcucConditionFormula`.`ecucQuery` **always required** ⌈The attribute `EcucConditionFormula`.`ecucQuery` shall always be defined **when the ECU Configuration Parameter definition is complete**.

⌋*()*

[constr_3586] `EcucConditionFormula`.`ecucQueryString` **always required** ⌈The attribute `EcucConditionFormula`.`ecucQueryString` shall always be defined **when the ECU Configuration Parameter definition is complete**.

⌋*()*

[constr_3587] `EcucValidationCondition`.`validationFormula` **always required** ⌈The attribute `EcucValidationCondition`.`validationFormula` shall always be defined **when the ECU Configuration Parameter definition is complete**.

⌋*()*

[constr_3588] `EcucValueCollection`.`ecuExtract` **always required** ⌈The attribute `EcucValueCollection`.`ecuExtract` shall always be defined **at code generation time**.

⌋*()*

[constr_3589] `EcucModuleConfigurationValues`.`ecucDefEdition` **always required** ⌈The attribute `EcucModuleConfigurationValues`.`ecucDefEdition` shall always be defined **at code generation time**.

⌋*()*

[constr_3590] `EcucModuleConfigurationValues`.`implementationConfigVariant` **always required** ⌈The attribute `EcucModuleConfigurationValues`.`implementationConfigVariant` shall always be defined **at code generation time**.

⌋*()*

[constr_3591] `EcucModuleConfigurationValues`.`definition` **always required** ⌈The attribute `EcucModuleConfigurationValues`.`definition` shall always be defined **at code generation time**.

⌋*()*

[constr_3592] `EcucContainerValue`.`definition` **always required** ⌈The attribute `EcucContainerValue`.`definition` shall always be defined **at code generation time**.

⌋*()*

[constr_3593] `EcucParameterValue.definition` **always required** ⌈The attribute `EcucParameterValue.definition` shall always be defined **at code generation time**.

⌋*()*

[constr_3594] `EcucNumericalParamValue.value` **always required** ⌈The attribute `EcucNumericalParamValue.value` shall always be defined **at code generation time**.

⌋*()*

[constr_3595] `EcucTextualParamValue.value` **always required** ⌈The attribute `EcucTextualParamValue.value` shall always be defined **at code generation time**.

⌋*()*

[constr_3596] `EcucAddInfoParamValue.value` **always required** ⌈The attribute `EcucAddInfoParamValue.value` shall always be defined **at code generation time**.

⌋*()*

[constr_3597] `EcucAbstractReferenceValue.definition` **always required** ⌈The attribute `EcucAbstractReferenceValue.definition` shall always be defined **at code generation time**.

⌋*()*

[constr_3598] `EcucInstanceReferenceValue.value` **always required** ⌈The attribute `EcucInstanceReferenceValue.value` shall always be defined **at code generation time**.

⌋*()*

[constr_3599] `EcucReferenceValue.value` **always required** ⌈The attribute `EcucReferenceValue.value` shall always be defined **at code generation time**.

⌋*()*

[constr_5015] Multiplicity of `multiplicityConfigClass` ⌈The multiplicity of the attribute `EcucCommonAttributes.multiplicityConfigClass` shall not exceed 3.

⌋*()*

[constr_5059] Ordering of `MetaDataItem`**s of a** `MetaDataType` ⌈The `MetaDataItem`s of a `MetaDataType` shall be ordered according to their `MetaDataItemLength`. `MetaDataItem`s with greater `MetaDataItemLength` going first.

⌋*()*

[constr_5108] `CddModuleId` **range restriction** ⌈The range of `CddModuleId` is restricted to the value 255 and to the range of values 2048..4095.

⌋*()*

[constr_5325] Existence of `upperMultiplicityInfinite` and `upperMultiplicity` is mutually exclusive ⌈The existence of the elements `upperMultiplicityInfinite` and `upperMultiplicity` shall be mutually exclusive.

⌋*()*

[constr_5342] `EcucDefinitionElement.upperMultiplicity` or `EcucDefinitionElement.upperMultiplicityInfinite` always required ⌈Exactly one of the attributes `EcucDefinitionElement.upperMultiplicity` or `EcucDefinitionElement.upperMultiplicityInfinite` shall always be defined **when the ECU Configuration Parameter definition is complete**.

⌋*()*

[constr_5345] Restriction for a reference destination in case of multiple aggregated `EcucParamConfContainerDef`s ⌈An `EcucReferenceDef` or `EcucChoiceReferenceDef` is not allowed to reference an `EcucParamConfContainerDef` as destination if

- this `EcucParamConfContainerDef` is aggregated by several `EcucParamConfContainerDef`s as `subContainer` and

- the `EcucParamConfContainerDef` structures in which the referenced `EcucParamConfContainerDef` is aggregated are different compared to the `EcucParamConfContainerDef` structure in which the `EcucReferenceDef` or `EcucChoiceReferenceDef` is located in.

⌋*()*

[constr_5500] Applicability of the `multiplicityConfigClass` attribute ⌈The `multiplicityConfigClass` attribute is applicable only to `EcucContainerDef`s which have `upperMultiplicity` greater than `lowerMultiplicity`.

⌋*()*

[constr_5502] Introduction of new `EcucParameterValue`s of type `EcucFunctionNameDef` at post-build time ⌈In case a new `EcucParameterValue`s of type `EcucFunctionNameDef` (see [TPS_ECUC_02033]) is introduced at post-build time, it's value shall be one of the existing function names (e.g. callouts). This means that it is not allowed to introduce new functions at post-build time.

⌋*()*

[constr_5504] Removing an instance of the `EcucContainerDef` at post-build time ⌈Only instances of `EcucContainerDef`s with `multiplicityConfigClass.configClass` set to `PostBuild` in the `multiplicityConfigClass.configVariant VariantPostBuild` which are not referenced or are exclusively referenced by `EcucAbstractReferenceDef`s with `valueConfigClass.configClass` set to `PostBuild` in the `valueConfigClass.configVariant VariantPostBuild` and

have been introduced at post-build time (not part of the initial configuration before post-build updates) can be removed at post-build time.

⌋*()*

[constr_5505] Configuration class of the elements of the `EcucQueryExpression` ⌈The elements of the `EcucQueryExpression` involved in one calculation formula shall have lower or equal configuration class (where `PreCompile` configuration class is considered to be the lowest and `PostBuild` the highest) with respect to the context element in which the calculation is performed (e.g. a `Link` configuration parameter can not calculate its value based on a `PostBuild` parameters value).

⌋*()*

[constr_5506] Applicability of `postBuildVariantMultiplicity` attribute ⌈The `postBuildVariantMultiplicity` attribute of `EcucContainerDef` is applicable only to `EcucContainerDef`s which have `upperMultiplicity` greater than `lowerMultiplicity`.

⌋*()*

[constr_5507] Value of `EcucContainerDef.postBuildVariantMultiplicity` if `postBuildVariantSupport` is set to `false` ⌈If `postBuildVariantSupport` is set to `false`, every `EcucContainerDef` in this `EcucModuleDef` with `upperMultiplicity` greater than `lowerMultiplicity` shall have its `postBuildVariantMultiplicity` attribute set to `false`.

⌋*()*

[constr_5508] Applicability of `postBuildVariantMultiplicity` attribute ⌈The `postBuildVariantMultiplicity` attribute is applicable only to `EcucCommonAttributes` which have `upperMultiplicity` greater than `lowerMultiplicity`.

⌋*()*

[constr_5509] Value of `postBuildVariantMultiplicity` if `postBuildVariantSupport` is set to `false` ⌈If `postBuildVariantSupport` is set to `false`, every `EcucCommonAttributes` in this `EcucModuleDef` with `upperMultiplicity` greater than `lowerMultiplicity` shall have its `postBuildVariantMultiplicity` attribute set to `false`.

⌋*()*

[constr_5510] Value of `postBuildVariantValue` if `postBuildVariantSupport` is set to `false` ⌈If `postBuildVariantSupport` is set to `false`, every `EcucCommonAttributes` in this `EcucModuleDef` shall have its `postBuildVariantValue` attribute set to `false`.

⌋*()*

[constr_5512] `postBuildVariantValue` **attribute of** `symbolicNameValue` **parameters** ⌈The values of `EcucParameterDef`s with `symbolicNameValue` attribute set to `true` shall have their `postBuildVariantValue` set to `false`.

⌋*()*

[constr_5514] Applicability of the `multiplicityConfigClass` **attribute** ⌈The `multiplicityConfigClass` attribute is applicable only to `EcucCommonAttributes` which have `upperMultiplicity` greater than `lowerMultiplicity`.

⌋*()*

[constr_5520] `valueConfigClass` **attribute of** `symbolicNameValue` **parameters** ⌈The values of `EcucParameterDef`s with `symbolicNameValue` attribute set to `true` shall have their `valueConfigClass`.`configClass` set to `PreCompile` or `PublishedInformation` for all `valueConfigClass`.`configVariant`s.

⌋*()*

[constr_5521] `multiplicityConfigClass` **attribute of** `symbolicNameValue` **parameters** ⌈The values of `EcucParameterDef`s with `symbolicNameValue` attribute set to `true` shall have their `multiplicityConfigClass`.`configClass` set to `PreCompile` for all `multiplicityConfigClass`.`configVariant`s.

⌋*()*

[constr_5522] `postBuildVariantMultiplicity` **attribute of** `symbolicNameValue` **parameters** ⌈The values of `EcucParameterDef`s with `symbolicNameValue` attribute set to `true` shall have their `postBuildVariantMultiplicity` set to `false`.

⌋*()*

[constr_5523] Allowed `configClass`**es for paired** `configVariant`**s** ⌈`PublishedInformation configClass` is supported by all `configVariant`s where [TPS_ECUC_02071] applies. Additionally, `VariantPreCompile configVariant` supports `PreCompile configClass`, `VariantLinkTime configVariant` supports `PreCompile` and `Link configClass`es, and `VariantPostBuild configVariant` supports `PreCompile`, `Link` and `PostBuild configClass`es.

⌋*()*

2.4 TPS_ECUResourceTemplate

[constr_3500] `category` **of** `HwAttributeDef` **shall not be extended** ⌈In contrast to the general rule that `category` can be extended by user-specific values it is **not allowed** to extend the meaning of the attribute `category` of meta-class `HwAttributeDef`

⌋*()*

[constr_3511] `HwType` shall not have a reference to another `HwType` ⌈A `HwType` (being a `HwDescriptionEntity`) shall not have a reference to another `HwType` in the role `hwType`. The definition of `HwType`s is not hierarchical.

⌋*()*

[constr_3512] No support of multiple instantiation ⌈An essential constraint is that each `HwElement` can only be target of one `nestedElement` reference. This means that there is no concept of multiple instantiation of hardware elements. If the same hardware element shall be used several times (using the `nestedElement` reference) each occurrence has to have its own description. This is also true for nested elements of the referenced nested element.

⌋*()*

[constr_3513] Scope of connections ⌈Each hardware connection shall only connect features which both are in the hierarchical scope of the hardware element. The hierarchical scope encloses

- all features belonging to the hardware element containing the connection

- all features belonging to hardware elements which are referenced directly and indirectly in the `nestedElement` relation from the hardware element containing connection.

⌋*()*

[constr_11001] Multiplicity of `HwAttributeValue.hwAttributeDef` ⌈For each `HwAttributeValue` the reference in the role `hwAttributeDef` shall exist.

⌋*()*

[constr_11002] Multiplicity of `HwElementConnector.hwElement` ⌈For each `HwElementConnector` there shall exist exactly 2 references in the role `hwElement`.

⌋*()*

[constr_11003] Multiplicity of `HwPinGroupConnector.hwPinGroup` ⌈For each `HwPinGroupConnector` there shall exist exactly 2 references in the role `hwPinGroup`.

⌋*()*

[constr_11004] Multiplicity of `HwPinConnector.hwPin` ⌈For each `HwPinConnector` there shall exist exactly 2 references in the role `hwPin`.

⌋*()*

[constr_11005] Multiplicity of `HwAttributeDef.isRequired` ⌈For each `HwAttributeDef` the attribute `isRequired` shall exist.

⌋*()*

2.5 TPS_GenericStructureTemplate

[constr_2501] Blueprint of blueprints are not supported ⌈Note that objects modeled particularly as a "blueprint" (e.g. `PortPrototypeBlueprint`) also live in a package of category `BLUEPRINT`. Strictly speaking this means that they can be "blueprints" of "blueprints". This indirection is not intended and not supported.

⌋*()*

[constr_2502] Merged model shall be compliant to the meta-model ⌈A model merged from ≪`atpSplitable`≫ elements shall adhere to the consistency rules of the *meta-model*. Note that the required lower multiplicities depend on the process phase therefore the AUTOSAR schema sets them mainly to 0. This also applies to the bound model.

⌋*()*

[constr_2504] Constraint to `bindingTime` ⌈The tag `vh.latestBindingTime` *constraints* the value of the attribute `bindingTime` from [TPS_GST_00190]. Hence, it defines the latest point in methodology which is allowed as value for `bindingTime` of this particular application of ≪`atpVariation`≫.

⌋*()*

[constr_2505] Multiplicity after binding ⌈*if* Phase $\geq$ {partRole}.BindingTime *then* number of {partRole}'s = n

⌋*()*

[constr_2507] `EvaluatedVariantSet` shall not refer to itself ⌈An `EvaluatedVariantSet` shall not refer to itself directly or via other `EvaluatedVariantSet`.

⌋*()*

[constr_2508] The `shortName` shall be unique in its name space ⌈The content of `shortName` needs to be unique (case insensitive) within a the parent given name space. Note that the check for uniqueness of `shortName` shall be performed case insensitively. This supports the good practice that names should not differ in upper / lower case only which would cause a lot of confusion.

The term "case insensitive" indicates that the characters in the sets

```
{a b c d e f g h i j k l m n o p q r s t u v w x y z}
{A B C D E F G H I J K L M N O P Q R S T U V W X Y Z}
```

are respectively considered to be the same. In other words case-insensitive check for uniqueness of `shortName`s results in the fact that e.g. elements with `shortName` "X" and "x" are considered the same and shall not exist in the same name space.

⌋*()*

[constr_2509] Uniqueness of `ReferenceBase.shortLabel` in the scope of an `ARPackage` ⌈The `shortLabel` of any given `ReferenceBase` defined in the scope of an `ARPackage` shall be unique within the scope of the enclosing `ARPackage`.

⌋*()*

[constr_2510] Only one default `ReferenceBase` ⌈Only one `ReferenceBase` per `ARPackage` can be defined as default (`isDefault`=true).

⌋*()*

[constr_2511] Named reference bases shall be available ⌈If there is a relative references, then one of the containing packages shall have a `referenceBase` with a `shortLabel` equal to the `base` of the reference.

⌋*()*

[constr_2512] `shortName` uniqueness constraint for variants ⌈`shortName` + `shortLabel` of a variant element shall be unique within the name space established by the surrounding `Identifiable`.

⌋*()*

[constr_2514] `shortLabel` in `VariationPoint` shall be unique ⌈The combination of `shortName` and `shortLabel` shall be unique within the next enclosing `Identifiable` {WholeClass}. In case the `shortName` does not exist on the {PartClass} the `shortLabel` is unnecessary. In case the `shortName` of the {PartClass} is unique in the context of the {WholeClass} the `shortLabel` is unnecessary.

⌋*()*

[constr_2515] Categories of packages shall not conflict ⌈

parent category	child + category (also indirect children)						
	empty	BLUEPRINT	STANDARD	EXAMPLE	ICS	custom1	custom2
empty	ok	ok	ok	ok	ok	ok	ok
BLUEPRINT	ok	ok	conflict	conflict	conflict	conflict	conflict
STANDARD	ok	conflict	ok	conflict	conflict	conflict	conflict
EXAMPLE	ok	conflict	conflict	ok	conflict	conflict	conflict
ICS	ok	conflict	conflict	conflict	ok	conflict	conflict
custom1	ok	conflict	conflict	conflict	conflict	ok	conflict
custom2	ok	conflict	conflict	conflict	conflict	conflict	ok

	target package category (if category is empty, then the parent category applies)						
category of package that contains reference source element (if category is empty, then the parent category applies)	empty	BLUEPRINT	STANDARD	EXAMPLE	ICS	cus-tom1	cus-tom2
empty	ok	ok	ok	ok	ok	ok	ok
BLUEPRINT	ok	ok	ok	conflict	ok	conflict	conflict
STANDARD	ok	conflict	ok	conflict	conflict	conflict	conflict
EXAMPLE	ok	ok	ok	ok	ok	conflict	conflict
ICS	ok	conflict	ok	conflict [4]	ok	conflict	conflict
custom1	ok	ok	ok	ok	ok	ok	ok
custom2	ok	ok	ok	ok	ok	ok	ok

⌋*()*

[constr_2516] Return type of an `AttributeValueVariationPoint` ⌈When such a formula is evaluated by a software tool, and the return value of the formula is shall be compatible to the type of the attribute in the pure meta-model.

⌋*()*

[constr_2517] `postBuildVariantCondition` only for PostBuild ⌈Aggregation of `PostBuildVariantCondition` in `VariationPoint` is only allowed if the annotated model states `vh.latestBindingTime` to `PostBuild`.

⌋*()*

[constr_2518] Binding time is constrained ⌈Note that this binding time is again constrained by the value of the tag `vh.latestBindingTime`.

⌋*()*

[constr_2519] `PredefinedVariant`s need to be consistent ⌈If a `Predefined-Variant` plus its `includedVariant`s references more than one `SwSystemcon-stantValueSet` all `value` attributes in `SwSystemconstValue`s for a particular `SwSystemconst` shall be identical.

⌋*()*

---

[4]see [constr_2573] for details

[constr_2520] Nesting of lists shall be limited ⌈The nesting of lists shall be limited to a reasonable depth such that it can safely be rendered on A4 pages. A reasonable approach is not to nest more than three levels.

⌋*()*

[constr_2521] The `shortLabel` in `AttributeValueVariationPoint` shall be unique ⌈The `shortLabel` shall be unique (case insensitive) within the next enclosing `Identifiable` and is used to individually address variation points in the *variant-rich M1 model*.

Note that the check for uniqueness of `shortLabel` shall be performed case insensitively. This supports the good practice that `shortLabel`s should not differ in upper / lower case only which would cause a lot of confusion.

The term 'case insensitive' indicates that the characters in the sets

```
{a b c d e f g h i j k l m n o p q r s t u v w x y z}
{A B C D E F G H I J K L M N O P Q R S T U V W X Y Z}
```

are respectively considered to be the same. In other words case-insensitive check for uniqueness of `shortLabel` results in the fact that e.g. elements with `shortLabel` 'X' and 'x' are considered the same and shall not exist in the same context.

⌋*()*

[constr_2522] Notes should not be nested ⌈Note even if it is possible to nest notes it is not recommended to do so, since it might lead to problems with the rendering of the note icon.

⌋*()*

[constr_2523] Used languages need to be consistent ⌈The used languages of an AUTOSAR file are specified in the top level `adminData`. All other elements shall be provided in the languages specified for the document.

⌋*()*

[constr_2524] Non splitable elements in one file ⌈If the *aggregation/attribute* is **not** ≪`atpSplitable`≫, then all aggregated element(s) shall be described in the same physical file as the aggregating element.

⌋*()*

[constr_2525] Non splitable elements shall not be repeated ⌈Properties (namely aggregations, references and primitive attributes) which are not marked as ≪`atp Splitable`≫ shall be placed in one physical file. They shall not be repeated in the split files unless they are an attribute which is used as a part of the split key. Another special case is handling of ≪`atpStructuredComment`≫, see [TPS_GST_00382].

⌋*()*

[constr_2533] Documentation context is either a feature or an identifiable ⌈One particular `DocumentationContext` shall be either a feature or an identifiable but not both at the same time. If this is desired, one should create multiple `Documentation-Context`.

⌋*()*

[constr_2534] Practically `UnlimitedInteger` shall be limited such that it fits into 64 bit. ⌈If a signed value is represented the min value can be down to -9223372036854775808 (0x8000000000000014) and the max value can be up to 9223372036854775807 (`0x7fffffffffffffff`).

⌋*()*

[constr_2537] Variation of `PackageableElement` is limited to components resp. modules ⌈Variation of `ARElement` in `ARPackage` shall be applied only to elements on a kind of component level. In particular this is `BswModuleDescription`, `Documentation`, `Implementation`, `SwComponentType`, `TimingExtension`. This constraint only applies if the `PackageableElement` is not a blueprint.

⌋*()*

[constr_2538] Global reference is limited to certain elements ⌈The ability to perform a global reference is limited to

- `Chapter`,
- `Topic1`,
- `Caption`,
- `Traceable`,
- `XrefTarget`,
- `Std`,
- `Xdoc`,
- `Xfile`

⌋*()*

[constr_2547] Ordered collections cannot be split into different partial models ⌈Ordered collections cannot be split. In other words: Contrary to the semantics of unordered collections - which can be distributed between partial models - ordered collections can only be placed as a whole in one of the partial models. Otherwise the merge approach would influence the semantics of the collections.

⌋*()*

[constr_2557] No `VariationPoint`s where `vh.latestBindingTime` set to `BlueprintDerivationTime` in system configurations ⌈Blueprints are **not** part of a system configuration. In consequence of this, in a system configuration there shall be

no `VariationPoint` where `vh.latestBindingTime` is restricted to `Blueprint DerivationTime` by the meta-model.

⌋*()*

[constr_2558] If `vh.latestBindingTime` is `BlueprintDerivationTime` then there shall only be `blueprintCondition` or `formalBlueprintGenerator` respectively `blueprintValue` ⌈`VariationPoint`s with `vh.latestBindingTime` restricted to `BlueprintDerivation` shall not have `swSyscond` nor `postBuild-VariantCondition`.

⌋*()*

[constr_2559] No nested `VariationPoint` ⌈As `blueprintCondition` is a `DocumentationBlock` it could again contain `VariationPoint`s and therefore would allow nesting of `VariationPoint`s. This is not intended and shall not be used.

⌋*()*

[constr_2567] Undefined Value in Attribute Value Blueprints ⌈If a `blueprint-Value` is specified, then the `value` defined by the `AttributeValueVariation-Point` is not used and should therefore at least contain one term `undefined` which is to be refined when deriving objects from this blueprint.

⌋*()*

[constr_2572] Unique Control of Document Languages ⌈The settings for multiple languages are specified in the top-Level `AdminData` only

⌋*()*

[constr_2573] ICS shall not reference examples ⌈ICS is like a productive Model and therefore shall not reference to an `EXAMPLE`. Such a reference would be useless since the target needs to be ignored in the ICS.

⌋*()*

[constr_2575] `blueprintValue` in blueprints only ⌈`blueprintValue` is only allowed in blueprints and may not be present in a system description.

⌋*()*

[constr_2577] Binding Time in Aggregation Pattern ⌈Within `VariationPoint`, the class `ConditionByFormula` has an attribute `bindingTime` which defines the *latest* binding time for this variation point. This binding time is further constrained by the UML tag `vh.latestBindingTime` that is attached to the aggregation see [TPS_GST_-00190], [TPS_GST_00220], [TPS_GST_00221]):

$$\text{ConditionByFormula.bindingTime} \leq \textit{aggregation}.\text{vh.latestBindingTime}$$

⌋*()*

[constr_2578] Binding Time in Association Pattern ⌈Within `VariationPoint`, the class `ConditionByFormula` has an attribute `bindingTime` which defines the *latest*

binding time for this variation point. This binding time is further constrained by the UML tag `vh.latestBindingTime` that is attached to the association (see [TPS_GST_-00190], [TPS_GST_00220],[TPS_GST_00221]):

$ConditionByFormula.bindingTime \leq association.vh.latestBindingTime$

⌋*()*

[constr_2579] Binding Time in Attribute Value Pattern ⌈The meta-class `AttributeValueVariationPoint` has an attribute `bindingTime` which defines the *latest* binding time for this variation point. This binding time is further constrained by the UML tag `vh.latestBindingTime` that is attached to the attribute (see [TPS_-GST_00190], [TPS_GST_00220], [TPS_GST_00221]):

$AttributeValueVariationPoint.bindingTime \leq attribute.vh.latestBindingTime$

⌋*()*

[constr_2580] Binding Time in Property Set Pattern ⌈The meta-class `VariationPoint` has an attribute `bindingTime` which defines the *latest* binding time for this variation point. This binding time is further constrained by the UML tag `vh.-latestBindingTime` that is attached to the meta-class which is marked as ≪atp Variation≫ (see [TPS_GST_00190], [TPS_GST_00220], [TPS_GST_00221]):

$VariationPoint.bindingTime \leq meta\ class.vh.latestBindingTime$

⌋*()*

[constr_2581] Default life cycle state shall be defined properly ⌈`defaultLcState` in `LifeCycleInfoSet` shall reference to a `lcState` defined in the `LifeCycleStateDefinitionGroup` referenced by `usedLifeCycleStateDefinitionGroup`.

⌋*()*

[constr_2583] Used life cycle state shall be defined properly ⌈`defaultLcState` in `LifeCycleInfo` shall reference to a `lcState` defined in the `LifeCycleStateDefinitionGroup` referenced by `usedLifeCycleStateDefinitionGroup` of the containing `LifeCycleInfoSet`.

⌋*()*

[constr_2585] `LifeCycleInfo` shall be unambiguous ⌈Within one particular `LifeCycleInfoSet lifeCycleInfo.lcObject` shall be unique. This ensures that the association of a `LifeCycleState` to a `Referrable` is unambiguous.

This contraint applies for a particular point in time under consideration of the period of viability according to [TPS_GST_00244].

⌋*()*

[constr_2586] Constraints on `LifeCyclePeriod` ⌈The attributes `date`, `arRelease-Version`, `productRelease` in `LifeCyclePeriod` are mutually exclusive.

⌋*()*

[constr_2587] No `System` in `AnyInstanceRef` ⌈In consequence of [TPS_GST_-00387] `System` shall not be `contextElement` nor `target` of an `AnyInstanceRef`. Otherwise `atpBase` would not be determined.

⌋*()*

[constr_2594] Cyclic value assignments to `SwSystemconst` is not allowed ⌈It is explicitly forbidden to assign values to `SwSystemconst` which in turn depend directly or indirectly on this value assignment.

⌋*()*

[constr_2595] Footnotes should not be nested ⌈Note that even if supported by the meta-model, footnotes shall not be nested. Nested footnotes might lead to problems with the processing of the footnote link. In other words `LParagraph` shall not be aggregated with role `ft` within a `LParagraph` which already has the role `ft`.

⌋*()*

[constr_2596] Used colors of attributes color and bgcolor ⌈The used colors of the attributes color and bgcolor shall base on the 6 digits RGB hex-code following

```
|#([a-fA-F0-9]{6})|
```

.

⌋*()*

[constr_2599] Maximum one VariationPoints in ≪`atpMixed`≫ ⌈In case an ≪atp Mixed≫ meta-class is aggregated as ≪atpVariation≫ there shall not be more than one `VariationPoint` and the `VariationPoint` shall be the last aggregated element.

⌋*()*

[constr_2601] Value of `AbstractEnumerationValueVariationPoint` ⌈The formula of an `AbstractEnumerationValueVariationPoint` shall evaluate to a value for which a mapping is defined in the `EnumerationMappingTable` which is referenced by the attributes `base` and `enumTable`.

⌋*()*

[constr_2602] Completeness of `AnyInstanceRef` referencing `Implementation-DataTypeElement` ⌈If the `target` references an `Implementation-DataTypeElement` the `AnyInstanceRef` shall define a `contextElement` reference for

1. each leaf `ImplementationDataTypeElement` in a chain of referencing `ImplementationDataType`s which is not the `target`

2. and each `ImplementationDataTypeElement` of category ARRAY in a chain of referencing `ImplementationDataType`s

Thereby the contexts are created according [TPS_GST_00162] from the root to the leaf `ImplementationDataTypeElement` which is either typed (directly or indirectly via `ImplementationDataType` of category TYPE_REFERENCE) or owns the `target`.

⌋*()*

[constr_2606] Existence of `upperMultiplicityInfinite` and `upperMultiplicity` of `AbstractMultiplicityRestriction` is mutually exclusive ⌈The existence of the elements `upperMultiplicityInfinite` and `upperMultiplicity` of `AbstractMultiplicityRestriction` shall be mutually exclusive.

⌋*()*

[constr_2607] `lowerMultiplicity` of `AbstractMultiplicityRestriction` shall be smaller or equal to `upperMultiplicity` ⌈`lowerMultiplicity` of `AbstractMultiplicityRestriction` shall be smaller or equal to `upperMultiplicity`.

⌋*()*

[constr_2626] `atpTarget` of InstanceRefs shall be consistent ⌈The `atpTarget` of an instance ref shall either

- be an `atpFeature` owned by the `atpType` of the last `atpContextElement` or

- be an `atpFeature` owned by an `AtpStructureElement` owned by the `atpType` of the last `atpContextElement`.

⌋*()*

[constr_2627] No reassigning of the same name within one LET Block ⌈Within one `LET` block one name shall be assigned to an value at most once.

⌋*()*

[constr_2628] Representation of xml.xsd.type=double data types ⌈All data types with xml.xsd.type=double shall comply with IEEE 754 and are limited to what can be expressed by a 64 bit binary representation.

⌋*()*

[constr_2629] Defined identity up to the root ⌈If an element in the M1 model aggregates splitable elements on deeper levels, it shall have a defined identity, i.e. the identifying attributes (e.g. `shortName` or `shortLabel`); see [TPS_GST_00047]; shall be set in the M1 model.

⌋*()*

[constr_2630] M1 elements with same identity but different type are not allowed
⌈Splitable M1 elements with the same identity but different type shall not exist.
⌋*()*

[constr_2631] Usage of value ANY for `AnyServiceInstanceId` ⌈The value of a given `AnyServiceInstanceId` shall not be set to ANY.
⌋*()*

[constr_2632] No postbuild variation for attribute values ⌈The tag `vh.latestBindingTime` is limited to `preCompileTime` and earlier binding times, i.e. (`blueprintDerivationTime`, `systemDesignTime` and `codeGenerationTime`) in the Attribute Value pattern.
⌋*()*

[constr_2633] Existence of reference decorated with stereotype ≪`isOfType`≫ ⌈If a subclass of `AtpPrototype` defines a reference decorated with stereotype ≪isOfType≫ to a subclass of `AtpType`, then this reference shall always exist.
⌋*()*

[constr_2634] Conditionals with ordered collections ⌈Ordered collections shall not be split over different conditionals.
⌋*()*

[constr_2635] No custom values for `Collection.category` ⌈It is not allowed to define any custom or project-specific value of the attribute `Collection.category`.
⌋*()*

[constr_2636] No custom values for `Collection.elementRole` ⌈It is not allowed to define any custom or project-specific value of the attribute `Collection.elementRole`.
⌋*()*

[constr_2637] Limits of `PositiveUnlimitedInteger` ⌈For `PositiveUnlimitedInteger`, the min value can be down to 0 and the max value can be up to 18446744073709551615 (`0xffffffffffffffff`).
⌋*()*

[constr_2638] Variation points shall not exists in non-variant roles ⌈Variation Points shall not exist if the aggregation is not stereotyped ≪atpVariation≫ in the AUTOSAR meta model at any time in the workflow.
⌋*()*

[constr_4055] ICS may not contain blueprints ⌈Since an Implementation Conformance Statement always describes a set of one or more fully configured software

modules, a package with category `ICS` it is not allowed to contain sub-packages at any level which have the category `BLUEPRINT`.

⌋*()*

2.6   TPS_SoftwareComponentTemplate

[constr_1000] End-to-end protection is limited to sender/receiver communication
⌈A `VariableDataPrototype` referenced in the roles

- `EndToEndProtectionVariablePrototype`.sender

- `EndToEndProtectionVariablePrototype`.receiver

shall be aggregated in the role `dataElement` at a `SenderReceiverInterface` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1004]   Mapping of `ApplicationDataType`s in the scope of single `AtomicSwComponentType`s ⌈In the scope of `AtomicSwComponentType`.internalBehavior.dataTypeMapping, each `ApplicationDataType` shall be mapped to exactly one `ImplementationDataType` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1005] Compatibility of `ImplementationDataType`s mapped to the same `ApplicationDataType` ⌈It is required that `ImplementationDataType`s which are taken for connecting corresponding elements of `PortInterface`s and thus refer to compatible `ApplicationDataType`s are also compatible among each other (so that RTE is able to cope with possible connections by converting the data accordingly) **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1006] Applicable data categories, depending on specific model elements related to data definition properties ⌈

Category	ApplicationArrayDataType	ApplicationRecordDataType	ApplicationPrimitiveDataType	ApplicationRecordElement	ApplicationArrayElement	ApplicationValueSpecification	ApplicationRuleBasedValueSpecification	ImplementationDataType	ImplementationDataTypeElement	SwSystemconst	McDataInstance	Calibration	Measurement	Communication Port Interfaces	RTE + BSW	Description
VALUE			x	x	x	x		x	x	x	x	x	x	x	x	Contains a single value.
VAL_BLK			x	x	x	x	x				x	x		x		A value block defines values stored together within one calibration parameter object. It is similar to an value array but it stores the values by means of an axis instead (only important for calibration data handling).
DATA_REFERENCE								x	x	x				x⁵	x	Contains an address of another `DataPrototype` (whose type is given via `SwDataDefProps.swPointerTargetProps`).
FUNCTION_REFERENCE								x	x	x					x	Contains an address of a function prototype (whose signature is given via `SwDataDefProps.swPointerTargetProps.functionPointerSignature`).
TYPE_REFERENCE								x	x	x				x	x	The element is defined via reference to another data type (via `SwDataDefProps.implementationDataType`).
STRUCTURE		x	x	x				x	x		x	x	x	x	x	Holds one or several further elements which can have different `AutosarDataType`s. The underlying elements are defined in the same manner as normal data except for the association to `SwAddrMethod`: This has to be the same for all underlying elements. Corresponds to a Record if used in the application domain.
UNION								x	x		x	x	x	x	x	Can hold values of different data types. It is similar to `STRUCTURE` except that all of its members start at the same location in memory. A `UNION` data prototype can contain only one of its elements at a time. The size of the `UNION` is at least the size of the largest member. Please find more information in [TPS_SWCT_01700].
ARRAY	x		x	x		x	x	x			x	x	x	x	x	An array of sub-elements which are of the same type.
BIT											x	x	x		x	One or several bits within a host variable, which are treated as an own data object.

▽

---

⁵[constr_1295] applies!

Category	ApplicationArrayDataType	ApplicationRecordDataType	ApplicationPrimitiveDataType	ApplicationRecordElement	ApplicationArrayElement	ApplicationValueSpecification	ApplicationRuleBasedValueSpecification	ImplementationDataType	ImplementationDataTypeElement	SwSystemconst	McDataInstance	Calibration	Measurement	Communication Port Interfaces	RTE + BSW	Description
HOST											x	x	x		x	A HOST data type is like a simple VALUE, but it is used for packed bit definition. That means it can host several BIT variables which have their own description and measurement access.
STRING	x	x	x	x							x	x	x	x		Contains a single value interpreted as a text string (note that it appears as a single value for the application domain; the internal representation can be an array).
BOOLEAN	x	x	x	x							x	x	x	x		Contains one boolean state. Depending on the CPU direct addressing of single bits may not be available. So a byte or a word can be used to store only one logical state.
COM_AXIS	x	x	x	x	x						x	x		x		An axis definition as separate calibration parameter which can be referenced by any CURVE, MAP, CUBOID, CUBE_4, and CUBE_5. The benefits by using a common axis is that it saves memory space; because it is stored only one time and can be used in multiple CURVEs, MAPs, CUBOIDs, CUBE_4s, and CUBE_5s.
RES_AXIS	x	x	x	x	x						x	x		x		A RES_AXIS (rescale axis) is also a shared axis like COM_-AXIS, the difference is that this kind of axis can be used for rescaling. Note that the RES_AXIS is by nature a CURVE which is used to implement a non linear scaling (rescale) of the axis. In addition to saving memory space via the shared usage like a COM_AXIS, it can compress a huge range to a non-linear distributed axis points thus retaining the required accuracy.
CURVE	x	x	x	x	x						x	x		x		Calibration parameter with one input value and one output value. That means output values can be defined depending on the input value. The granularity of implemented functionality can be changed by using different number of axis points. A CURVE has always one input axis and one output axis. The output axis is a characteristic of the curve and every time present but the input axis can be defined within the curve definition or separately.

△

Category	Applicable to ...										Use Case				Description
	ApplicationArrayDataType	ApplicationRecordDataType	ApplicationPrimitiveDataType	ApplicationRecordElement	ApplicationArrayElement	ApplicationValueSpecification	ApplicationRuleBasedValueSpecification	ImplementationDataType	ImplementationDataTypeElement	SwSystemconst	McDataInstance	Calibration	Measurement	Communication Port Interfaces	RTE + BSW	
MAP		x	x	x	x	x					x	x		x		Calibration parameter with two input values and one output value. That means output values can be defined depending on the input values. The granularity of implemented functionality can be changed by using different number of axis points for y- and x-axis. A MAP has always two input axes and one output axis. The output axis is a characteristic of the MAP and every time present but the input axes can be defined within the MAP definition or separately.
CUBOID		x	x	x	x	x					x	x		x		Calibration parameter with three input values and one output value. That means output values can be defined depending on the input values. The granularity of implemented functionality can be changed by using different number of axis points for the input axes. A CUBOID has always three input axes and one output axis. The output axis is a characteristic of the CUBOID and every time present but the input axes can be defined within the CUBOID definition or separately.
CUBE_4		x	x	x	x	x					x	x		x		Calibration parameter with four input values and one output value. That means output values can be defined depending on the input values. The granularity of implemented functionality can be changed by using different number of axis points for the input axes. A CUBE_4 has always four input axes and one output axis. The output axis is a characteristic of the CUBE_4 and every time present but the input axes can be defined within the CUBE_4 definition or separately.
CUBE_5		x	x	x	x	x					x	x		x		Calibration parameter with five input values and one output value. That means output values can be defined depending on the input values. The granularity of implemented functionality can be changed by using different number of axis points for the input axes. A CUBE_5 has always five input axes and one output axis. The output axis is a characteristic of the CUBE_5 and every time present but the input axes can be defined within the CUBE_5 definition or separately.

⌋()

[constr_1007] Allowed attributes of `SwDataDefProps` for `Application-DataType`s ⌐

Attributes of SwDataDefProps	ApplicationDataType	ApplicationRecordElement	ApplicationArrayElement	VALUE	VAL_BLK	STRUCTURE	ARRAY	STRING	BOOLEAN	COM_AXIS	RES_AXIS	CURVE	MAP	CUBOID	CUBE_4	CUBE_5
	Root Elem.			**Attribute Existence per Category**												
additionalNativeTypeQualifier																
annotation	x	x	x	*	*	*	*	*	*	*	*	*	*	*	*	*
baseType																
compuMethod	x			0..1	0..1				0..1			0..1	0..1	0..1	0..1	0..1
dataConstr.dataConstrRule.physConstrs	x	x	x	0..1	0..1		0..1		0..1			0..1	0..1	0..1	0..1	0..1
dataConstr.dataConstrRule.internalConstrs	x	x	x	d/c[6]	d/c		d/c		d/c			d/c	d/c	d/c	d/c	d/c
displayFormat	x	x	x	0..1	0..1		0..1	0..1	0..1			0..1	0..1	0..1	0..1	0..1
displayPresentation	x	x	x	0..1	0..1		0..1			0..1	0..1	0..1	0..1	0..1	0..1	0..1
implementationDataType																
invalidValue	x			0..1				0..1	0..1							
stepSize	x	x	x	0..1	0..1		0..1			0..1	0..1	0..1	0..1	0..1	0..1	0..1
swAddrMethod	x			0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1
swAlignment																
swBitRepresentation																
swCalibrationAccess	x	x		0..1	0..1	0..1	0..1	0..1	0..1	1	1	1	1	1	1	1
swCalprmAxisSet	x									1	1	1	1	1	1	1
swComparisonVariable																
swDataDependency																
swHostVariable																
swImplPolicy	x			0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1
swIntendedResolution	x	x	x	0..1												
swInterpolationMethod	x			0..1						0..1	0..1	0..1	0..1	0..1	0..1	0..1
swIsVirtual																
swPointerTargetProps																
swRecordLayout	x			0..1	0..1[7]				0..1	1	1	1	1	1	1	1
swRefreshTiming	x			0..1	0..1				0..1	0..1						
swTextProps	x							1								
swValueBlockSize	x				1											
swValueBlockSizeMult	x				1											
unit	x			0..1	0..1			0..1	0..1			0..1	0..1	0..1	0..1	0..1
valueAxisDataType	x					0..1				0..1	0..1	0..1	0..1	0..1	0..1	0..1
Other Attributes below the Root Element																
element: ApplicationRecordElement	x	x	x			1..*										

▽

---

[6] don't care
[7] This is required by [TPS_SWCT_01179].

△

Attributes of SwDataDefProps	Root Elem.			Attribute Existence per Category													
	ApplicationDataType	ApplicationRecordElement	ApplicationArrayElement	VALUE	VAL_BLK	STRUCTURE	ARRAY	STRING	BOOLEAN	COM_AXIS	RES_AXIS	CURVE	MAP	CUBOID	CUBE_4	CUBE_5
element: ApplicationArrayElement	x	x	x				1									
ApplicationArrayElement.array-SizeSemantics	x						0..1									
ApplicationArrayElement. maxNumberOfElements	x						1									

⌋ ()

[constr_1009] SwDataDefProps applicable to ImplementationDataTypes ⌈

Attributes of SwDataDefProps	Root Element				Attribute Existence per Category						
	ImplementationDataType	ImplementationDataTypeElement	SwPointerTargetProps	SwServiceArg	VALUE	DATA_REFERENCE	FUNCTION_REFERENCE	TYPE_REFERENCE	STRUCTURE	UNION	ARRAY
additionalNativeTypeQualifier	x	x	x	x	0..1	0..1	0..1	0..1	0..1	0..1	0..1
annotation	x	x	x	x	*	*	*	*	*	*	*
baseType	x	x	x	x	1						
compuMethod	x	x	x	x	0..1			0..1			
dataConstr.dataConstrRule.physConstrs	x	x	x	x	d/c[8]			d/c			d/c
dataConstr.dataConstrRule.internalConstrs	x	x	x	x	0..1			0..1			0..1
displayFormat	x	x			0..1				0..1	0..1	0..1
displayPresentation	x	x			0..1						0..1
implementationDataType	x	x	x	x				1			

▽

---

[8]don't care

△

Attributes of SwDataDefProps	Root Element				Attribute Existence per Category						
	ImplementationDataType	ImplementationDataTypeElement	SwPointerTargetProps	SwServiceArg	VALUE	DATA_REFERENCE	FUNCTION_REFERENCE	TYPE_REFERENCE	STRUCTURE	UNION	ARRAY
invalidValue	x	x	x		0..1			0..1	0..1[9]		0..1[10]
stepSize	x	x			0..1						
swAddrMethod	x	x	x		0..1	0..1	0..1	0..1	0..1	0..1	0..1
swAlignment	x				0..1	0..1	0..1		0..1	0..1	0..1
swBitRepresentation											
swCalibrationAccess	x	x			0..1			0..1	0..1	0..1	0..1
swCalprmAxisSet											
swComparisonVariable											
swDataDependency											
swHostVariable											
swImplPolicy	x		x	x	0..1	0..1	0..1	0..1	0..1	0..1	0..1
swIntendedResolution											
swInterpolationMethod											
swIsVirtual											
swPointerTargetProps	x	x	x	x		1	1				
swPointerTargetProps.swDataDefProps	x	x	x	x		1					
swPointerTargetProps.functionPointerSignature	x	x	x	x			1				
swRecordLayout											
swRefreshTiming	x	x	x	x	0..1				0..1	0..1	0..1
swTextProps											
swValueBlockSize											
swValueBlockSizeMult											
unit											
valueAxisDataType											
Other Attributes											
subElement: ImplementationDataTypeElement	x	x							1..*	1..*	1
subElement.arraySizeSemantics	x	x									0..1
subElement.arraySize	x	x									1

⌋()

---

[9]There is a use case for the definition of an invalidValue for category ARRAY and therefore category STRUCTURE is also supported for the sake of symmetry.

[10]This represents an exception such that it would make sense to use an entire ArrayValueSpecification as the invalidValue because a string semantically is more than just a bunch of characters in a row.

[constr_1010] If `nativeDeclaration` does not exist ⌈If `nativeDeclaration` does not exist in the `SwBaseType`, it is required that the `shortName` (e.g. "uint8") of the corresponding `ImplementationDataType` is equal to a name of one of the Platform or Standard Types predefined in AUTOSAR code.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1011] `category` of `SwBaseType` ⌈For the attribute `SwBaseType.category` only the values `FIXED_LENGTH` and `VOID` are supported **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1012] Value of `category` is `FIXED_LENGTH` ⌈If the value of the attribute `SwBaseType.category` is set to `FIXED_LENGTH` then the attribute `baseTypeSize` shall be filled with content **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1015] Prioritization of `SwDataDefProps` ⌈

Attributes of SwDataDefProps	Usage For			Place of Setting										
	RTE	A2L	Other Usage	ApplicationDataType	ImplementationDataType	DataPrototype	InstantiationDataDefProps	ParameterAccess	SwServiceArg	FlatInstanceDescriptor	McDataInstance	SwSystemconst	PerInstanceMemory	ComSpec networkRepresentation
additionalNativeTypeQualifier	x		x	NA	D	I	NA	NA	D	NA	S	NA	NA	NA
annotation			x	D	A	A	A	A	D	NA	A	D	NA	A
baseType	x	x	x	NA	D	I	I	I	D	NA	S	M	NA	R
compuMethod	x	x	x	D	AI	I	I	NA	I	AI	S	D	NA	R
dataConstr	x	x	x	D	C	R	R	I	R	NA	S	D	NA	NA
displayFormat		x		D	A	R	R	I	R	NA	S	D	NA	NA
displayPresentation	x	x	x	D	A	R	R	NA	NA	NA	S	NA	NA	NA
implementationDataType	x		x	NA	D	I	I	I	D	NA	NA	NA	NA	NA
invalidValue	x	x		D	A	I	I	NA	NA	NA	S	NA	NA	D
stepSize		x		D	A	A	A	A	NA	A	S	NA	NA	NA
swAddrMethod	x	x	x	D	R	R	R	NA	NA	R	NA	NA	D	NA
swAlignment	x		x	NA	D	R	R	NA	NA	NA	NA	NA	NA	NA
swBitRepresentation		x	x	NA	NA	NA	NA	NA	NA	NA	D	NA	NA	NA

▽

△

Attributes of SwDataDefProps	Usage For			Place of Setting										
	RTE	A2L	Other Usage	ApplicationDataType	ImplementationDataType	DataPrototype	InstantiationDataDefProps	ParameterAccess	SwServiceArg	FlatInstanceDescriptor	McDataInstance	SwSystemconst	PerInstanceMemory	ComSpec networkRepresentation
swCalibrationAccess	x	x		D	R	R	R	NA	R	R	S	D	NA	NA
swCalprmAxisSet	x	x		D	NA	I	I	I	NA	NA	S	NA	NA	NA
swCalprmAxisSet.swCalprmAxis /SwAxisGrouped.swCalprmRef		x		NA	NA	NA	D	R	NA	NA	S	NA	NA	NA
swCalprmAxisSet.swCalprmAxis /SwAxisIndividual.swVariableRef		x		NA	NA	NA	D	R	NA	NA	S	NA	NA	NA
swCalprmAxisSet.swCalprmAxis /SwAxisGrouped.sharedAxisType		x		D	NA	NA	NA	NA	NA	NA	S	NA	NA	NA
swCalprmAxisSet.swCalprmAxis /SwAxisIndividual.inputVariableType		x		D	NA	NA	NA	NA	NA	NA	S	NA	NA	NA
swCalprmAxisSet/SwAxisIndividual.unit		opt.		D	NA	I	I	I	I	NA	S	NA	NA	NA
swComparisonVariable		x		NA	NA	NA	NA	D	NA	NA	S	NA	NA	NA
swDataDependency		x	x	NA	NA	D	R	NA	NA	NA	S	NA	NA	NA
swHostVariable		x	x	NA	NA	NA	NA	NA	NA	NA	D	NA	NA	NA
swImplPolicy	x		x	D	A	A	NA	NA	D	NA	NA	NA	NA	NA
swIntendedResolution			x	D[11]	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
swInterpolationMethod			x	D	I	R	R	R	NA	NA	S	NA	NA	NA
swIsVirtual		x		NA	NA	D	R	NA	NA	NA	S	NA	NA	NA
swPointerTargetProps			x	NA	D	I	NA	NA	D	NA	NA	NA	NA	NA
swRecordLayout	x	x	x	D	NA	I	I	I	NA	NA	S	NA	NA	NA
swRefreshTiming		x		D	R	R	R	NA	R	R	R	NA	NA	NA
swTextProps		x	x	D	I	I	I	I	NA	NA	S	NA	NA	NA
swValueBlockSize		x	x	D	I	I	I	I	NA	NA	S	NA	NA	NA
swValueBlockSizeMult		x	x	D	I	I	I	I	NA	NA	S	NA	NA	NA
unit		x	x	D	I	I	I	NA	I	NA	S	D	NA	NA
valueAxisDataType		x	x	D	I	I	I	I	NA	NA	S	NA	NA	NA

⌋()

[constr_1016] Restriction of `invalidValue` for `ImplementationDataType` and `ImplementationDataTypeElement` ⌈invalidValue for Implementation‑DataType and ImplementationDataTypeElement is restricted to be either a compatible NumericalValueSpecification, TextValueSpecification (caution, [constr_1284] applies) or a ConstantReference that in turn points to a compatible ValueSpecification.

---

[11] swIntendedResolution is used only in an early phase of the definition of data types, namely in the context of the definition of so-called blueprints. To that extent, swIntendedResolution represents a non-binding requirement that shall later be considered for the definition of an appropriate CompuMethod.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1017] **Supported combinations of `swImplPolicy` and `swCalibrationAccess`** ⌈

swImplPolicy	swCalibrationAccess		
	notAccessible	readOnly	readWrite
fixed	**yes**	not supported	not supported
const	**yes**	**yes**	not supported
standard	**yes**	**yes**	**yes**
queued	**yes**	not supported	not supported
measurementPoint	not supported	**yes**	not supported

⌋*()*

[constr_1018] **`dataElement` with `swImplPolicy` set to `measurementPoint` shall not be referenced by a `VariableAccess` aggregated by `RunnableEntity` in the role `dataReadAccess`** ⌈Due to the nature of `dataElement`s characterized by setting the `swImplPolicy` to `measurementPoint`, such `dataElement`s shall not be referenced by a `VariableAccess` aggregated by `RunnableEntity` in the role `dataReadAccess` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1020] `ParameterDataPrototype` needs to be of compatible data type as referenced in `sharedAxisType` ⌈Finally, the `ParameterDataPrototype` assigned in `swCalprmRef` shall be typed by data type compatible to `sharedAxisType` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1022] Limits shall be defined for each direction of `CompuMethod` ⌈In case that both domains are specified in the `CompuMethod` both shall have explicitly defined limits **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1024] Stepwise definition of `CompuMethod`s ⌈In a bound model, the intervals (i.e. determined by attributes `CompuScale.lowerLimit` and `CompuScale.upperLimit`) defined by `CompuScale`s used in the context of a given `CompuMethod` of all values of `category` except `BITFIELD_TEXTTABLE` shall **not** overlap.

For `CompuMethod`s of `category` `BITFIELD_TEXTTABLE`, the combination of the interval created by attributes `CompuScale.upperLimit`, `CompuScale.lowerLimit` and `CompuScale.mask` shall be unique in the context of the enclosing `CompuMethod`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1025] Avoid division by zero in rational formula ⌈The rational formula shall not yield any division by zero **at any time in the workflow**.

⌋*()*

[constr_1026] Compatibility of `Units` ⌈If a `SwDataDefProps` references a `Unit` and the `SwDataDefProps` has a reference to either of/or both

- a `CompuMethod` that in turn references a `Unit`

- a `DataConstr` that in turn references a `Unit`

then the `Unit`s referenced from

- `SwDataDefProps`

- `SwDataDefProps`.`compuMethod`

- `SwDataDefProps`.`dataConstr`

shall be compatible **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1029] `ConstantSpecificationMapping` and `ConstantSpecification` ⌈It is required that one `ConstantSpecification` referenced from a `ConstantSpecificationMapping` needs to be defined in the application domain (`applConstant`) and the other referenced `ConstantSpecification` needs to be defined in the implementation domain (`implConstant`).

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1033] Communication scenarios for sender/receiver communication ⌈For sender/receiver communication, it is not allowed to create a communication scenario where n sender are connected to m receivers where m and n are **both** greater than 1 **at any time in the workflow**.

⌋*()*

[constr_1035] Recursive definition of `CompositionSwComponentType` ⌈The recursive definition of a `CompositionSwComponentType` that eventually contains a `SwComponentPrototype` typed by the same `CompositionSwComponentType` shall not be feasible **at any time in the workflow**.

⌋*()*

[constr_1036] Connect kinds of `PortInterface`s ⌈It shall not be possible to connect `PortPrototype`s typed by `PortInterface`s of different kinds **at the time when the RTE is generated**.

Subclasses of `DataInterface` make an exception to this rule and can be used for creating connections to each other.

⌋*()*

[constr_1037] Client shall not be connected to multiple servers ⌈A client shall not be connected to multiple servers such that an operation call would be handled by more than one server **at the time when the RTE is generated**.

⌋*()*

[constr_1038] Reference to `ApplicationError` ⌈A `possibleError` referenced by a `ClientServerOperation` shall be owned by the `ClientServerInterface` that also owns the `ClientServerOperation` **at any time in the workflow**.

⌋*()*

[constr_1039] Relevance of `swImplPolicy` ⌈It is not possible to define a mapping between an element where the `swImplPolicy` is set to `queued` and another element where the `swImplPolicy` is set differently. This rule shall be imposed **at any time in the workflow**.

⌋*()*

[constr_1040] Conversion of `SenderReceiverInterface`s ⌈The conversion of elements of `SenderReceiverInterface`s is possible if one of the following conditions applies **at the time when the RTE is generated**:

- The `AutosarDataType`s of the referred `DataPrototype`s are compatible.

- A conversion of the data is available.

- A `DataPrototypeMapping.firstToSecondDataTransformation` is defined.

⌋*()*

[constr_1041] Conversion of `ClientServerInterface`s ⌈Either the `AutosarDataType`s of the referred `ArgumentDataPrototype`s are compatible or a conversion of the data is available **at the time when the RTE is generated**.

⌋*()*

[constr_1043] Allowed combinations of a specific *Type of* `PortInterface`**, a specific** *Type of* `PortPrototype`**, and a specific** *Type of* `ComSpec` ⌈

Type of PortPrototype	Type of ComSpec	Role of Element	Type of PortInterface	Role of Type-Ref
PPortPrototype	NonqueuedSender-ComSpec	dataElement	SenderReceiverIn-terface	providedInterface
PPortPrototype	QueuedSenderCom-Spec	dataElement	SenderReceiverIn-terface	providedInterface
RPortPrototype	Nonqueue-dReceiverComSpec	dataElement	SenderReceiverIn-terface	requiredInterface
RPortPrototype	QueuedReceiver-ComSpec	dataElement	SenderReceiverIn-terface	requiredInterface
PRPortPrototype	NonqueuedSender-ComSpec	dataElement	SenderReceiverIn-terface	providedRequired-Interface
PRPortPrototype	Nonqueue-dReceiverComSpec	dataElement	SenderReceiverIn-terface	providedRequired-Interface
PRPortPrototype	QueuedReceiver-ComSpec	dataElement	SenderReceiverIn-terface	providedRequired-Interface
PRPortPrototype	QueuedSenderCom-Spec	dataElement	SenderReceiverIn-terface	providedRequired-Interface
PPortPrototype	NvProvideComSpec	nvData	NvDataInterface	providedInterface
RPortPrototype	NvRequireComSpec	nvData	NvDataInterface	requiredInterface
PRPortPrototype	NvProvideComSpec	nvData	NvDataInterface	providedRequired-Interface
PRPortPrototype	NvRequireComSpec	nvData	NvDataInterface	providedRequired-Interface
PPortPrototype	ModeSwitchSender-ComSpec	modeGroup	ModeSwitchInter-face	providedInterface
RPortPrototype	ModeSwitchRe-ceiverComSpec	modeGroup	ModeSwitchInter-face	requiredInterface
PRPortPrototype	ModeSwitchSender-ComSpec	modeGroup	ModeSwitchInter-face	providedRequired-Interface
PRPortPrototype	ModeSwitchRe-ceiverComSpec	modeGroup	ModeSwitchInter-face	providedRequired-Interface
PPortPrototype	ParameterProvide-ComSpec	parameter	ParameterInter-face	providedInterface
RPortPrototype	ParameterRequire-ComSpec	parameter	ParameterInter-face	requiredInterface
PPortPrototype	ServerComSpec	operation	ClientServerIn-terface	providedInterface
RPortPrototype	ClientComSpec	operation	ClientServerIn-terface	requiredInterface
PRPortPrototype	ServerComSpec	operation	ClientServerIn-terface	providedRequired-Interface
PRPortPrototype	ClientComSpec	operation	ClientServerIn-terface	providedRequired-Interface

⌋*()*

[constr_1044] Applicability of `DataFilter` ⌈According to the origin of `DataFilter`, i.e. ISO 17356-4 specification [6], `DataFilter`s can only be applied to values with an integer base type **at any time in the workflow**.

⌋*()*

[constr_1046] Applicability of [TPS_SWCT_01845] ⌈[TPS_SWCT_01845] applies **at the time when the contract phase generation is executed only** if the value of the attribute `isService` is set to `false`.

⌋*()*

[constr_1047] Compatibility of `ApplicationPrimitiveDataTypes` ⌈Instances of `ApplicationPrimitiveDataType` are compatible **at the time when the RTE is generated** if and only if one of the following conditions applies:

1. All the following sub conditions apply:

   (a) They have the same `category`.

   (b) The `swDataDefProps` (after consideration of [constr_1015]) attached to the M1 data types are compatible.

2. In the context of using the `ApplicationPrimitiveDataType`, a `DataPrototypeMapping` exists that refers to a `DataPrototype` typed by one of the `ApplicationPrimitiveDataType`s in the role `firstDataPrototype` and to another `DataPrototype` typed by the other `ApplicationPrimitiveDataType` in the role `secondDataPrototype`.

3. In the context of using the `ApplicationPrimitiveDataType`, a `DataPrototypeMapping` exists that refers to a `DataPrototype` typed by the `ApplicationPrimitiveDataType` in the role `secondDataPrototype` and to another `DataPrototype` typed by an `ApplicationCompositeDataType` in the role `firstDataPrototype` and additionally for the side of the `ApplicationCompositeDataType` a corresponding `ApplicationCompositeDataTypeSubElementRef` exists in the role `firstElement` that in turn references an `ApplicationCompositeElementDataPrototype`.

⌋*()*

[constr_1048] Compatibility of `ApplicationRecordDataTypes` ⌈Instances of `ApplicationRecordDataType`s are compatible **at the time when the RTE is generated** if and only if one of the following conditions applies:

1. All `element`s *at the same record position* are of compatible `AutosarDataType`s (either `ApplicationCompositeDataType`s or `ApplicationPrimitiveDataType`s).

2. For each `ApplicationRecordDataType.element`, the attribute `isOptional` shall either

   • not exist on both sides or

   • be set to the value `false` if it only exists on one side or

   • have the identical value on both sides.

3. In the context of a `DataPrototypeMapping`, for each `ApplicationRecordElement` of the required `ApplicationRecordDataType` a `SubEle-`

mentMapping exists such that a `ApplicationCompositeDataType-SubElementRef` in the role `firstElement` or `secondElement` exists that references the required `ApplicationRecordElement` **and** a corresponding `ApplicationCompositeDataTypeSubElementRef` exists in the **other** role (i.e. `secondElement` or `firstElement`) that in turn references an `ApplicationRecordElement` of the provided `ApplicationRecordDataType`.

⌋*()*

[constr_1049] Compatibility of `ApplicationArrayDataType`s ⌈Instances of `ApplicationArrayDataType` are compatible **at the time when the RTE is generated** if and only if one of the following conditions applies:

1. All the following sub conditions apply:

   (a) Their `element`s are of a compatible `AutosarDataType`s (either `ApplicationCompositeDataType`s or `ApplicationPrimitive-DataType`s).

   (b) The attributes `maxNumberOfElements` and `arraySizeSemantics` (given the existence) have identical values.

2. In the context of a `DataPrototypeMapping`, for the `ApplicationArrayElement` of the required `ApplicationArrayDataType` a `SubElementMapping` exists such that a `ApplicationCompositeDataTypeSubElementRef` in the role `firstElement` or `secondElement` exists that references the required `ApplicationArrayElement` **and** a corresponding `ApplicationCompositeDataTypeSubElementRef` exists in the **other** role (i.e. `secondElement` or `firstElement`) that in turn references an `ApplicationArrayElement` of the provided `ApplicationArrayDataType`.

⌋*()*

[constr_1050] Compatibility of `ImplementationDataType`s ⌈Instances of `ImplementationDataType` are compatible **at the time when the RTE is generated** if and only if after all type-references are resolved one of the following rules apply:

1. All the following sub conditions apply:

   (a) They have the same `category`.

   (b) They have the identical structure (this refers to `ImplementationDataTypeElement` and their `subElement`s).

   (c) The attributes `arraySize` and `arraySizeSemantics` have (given the existence) identical values.

   (d) For each `ImplementationDataType.subElement`, the attribute `isOptional` shall either

   • not exist on both sides or

- be set to the value `false` if it only exists on one side or

- have the identical value on both sides.

(e) The `swDataDefProps` (after consideration of [constr_1015]) attached to the M1 data types are compatible.

2. In the context of using the `ImplementationDataType`, a `DataPrototypeMapping` exists that refers to a `DataPrototype` typed by one of the `ImplementationDataType`s in the role `firstDataPrototype` and to another `DataPrototype` typed by the other `ImplementationDataType` in the role `secondDataPrototype`.

3. In the context of using the `ImplementationDataType`, a `DataPrototypeMapping` exists that refers to a `DataPrototype` typed by the `ImplementationDataType`s in the role `secondDataPrototype` and to another `DataPrototype` typed by an `ImplementationDataType` with a `subElement` in the role `firstDataPrototype` and additionally for the side of the `ImplementationDataType` with a `subElement` a corresponding `ImplementationDataTypeSubElementRef` exists in the role `firstElement` that in turn references an `ImplementationDataTypeElement`.

⌋*()*

[constr_1051] Compatibility of `SwDataDefProps` ⌈SwDataDefProps are compatible **at the time when the RTE is generated** if and only if:

1. They refer to compatible `Unit` definitions, or neither of them has an associated `Unit`.

2. They refer to compatible conversion methods or neither of them associates such a method.

3. They both aggregate a `ValueSpecification` in the role `invalidValue` or neither of them aggregates a `ValueSpecification` in the role `invalidValue`.

4. If existent (see previous condition), one of the following conditions apply to `ValueSpecification`s aggregated in the role `invalidValue` for being considered compatible (after following and resolving indirections created by `ConstantReference`):

(a) both are `ApplicationValueSpecification`s and the values are compatible according to [TPS_GST_02501].

(b) both are `NumericalValueSpecification`s and the values are compatible according to [TPS_GST_02501].

(c) both are `TextValueSpecification`s and the values are identical.

(d) both are `ArrayValueSpecification`s and the values are effectively identical, e.g. if one `ArrayValueSpecification` specifies all values ex-

plicitly and the other `ArrayValueSpecification` specifies values based on a rule then the yield of both `ArrayValueSpecification`s (i.e. element for element) shall be identical.

(e) both are `RecordValueSpecification`s and the values are identical.

(f) if one is a `NumericalValueSpecification` and the other one is an `ApplicationValueSpecification` then the check for compatibility shall apply the `CompuMethod` on the physical value such that a comparison on the implementation level becomes possible. [TPS_GST_02501] applies[12].

5. They refer to compatible data constraints `dataConstr`.

6. They refer to compatible `swRecordLayout`s

All other attributes (e.g. `swCalibrationAccess` do not affect compatibility).

⌋*()*

[constr_1052] Compatibility of `Units` ⌈Two `Unit` definitions are compatible **at the time when the RTE is generated** if and only if:

1. They have compatible (see [TPS_GST_02501]) values of attributes `factorSiToUnit` and `offsetSiToUnit`.

2. One of the following conditions is fulfilled:

    • They refer to compatible definitions of `PhysicalDimension`.

    • Neither of them associates a `PhysicalDimension`.

    • One `Unit` refers to a `PhysicalDimension` with `shortName` NoDimension where all exponents are set to 0 and the other `Unit` does not refer to a `PhysicalDimension`.

⌋*()*

[constr_1053] Compatibility of `PhysicalDimension`s ⌈Two `PhysicalDimension` definitions are compatible if and only if the values of

    • `lengthExp`

    • `massExp`

    • `timeExp`

    • `currentExp`

    • `temperatureExp`

    • `molarAmountExp`

---

[12]if one is a `NumericalValueSpecification` and the other one is an `ApplicationValueSpecification` and the application of the `CompuMethod` on the side of the `ApplicationValueSpecification` does not yield a valid number a comparison is not possible.

- luminousIntensityExp

are identical and **either**

- the shortNames are identical **or**

- a PhysicalDimensionMapping exists that maps one of the PhysicalDimensions in the role firstPhysicalDimension and the other PhysicalDimension in the role secondPhysicalDimension.

The imposition time of this constraint depends on the context:

- If the compatibility of PhysicalDimensions is evaluated in the context of the creation of a SwConnector, then the rule shall be imposed **at the time when the RTE is generated**.

- If the context is the creation of an ApplicationValueSpecification, then the rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1054] No DataConstr available at the provider ⌈If the provider defines no constraints, it is only compatible with a receiver which also defines no constraints at all **at the time when the RTE is generated**.

⌋*()*

[constr_1055] ImplementationDataType has category VALUE ⌈The attributes baseType shall refer to a compatible SwBaseType **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1056] ImplementationDataType has category TYPE_REFERENCE ⌈The ImplementationDataTypes referenced by the attributes SwDataDefProps.implementationDataType shall be compatible **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1057] ImplementationDataType has category DATA_REFERENCE ⌈The attributes SwDataDefProps.swPointerTargetProps shall have identical targetCategory and shall refer to SwDataDefProps where all attributes are identical **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1058] ImplementationDataType has category FUNCTION_REFERENCE ⌈The attributes SwDataDefProps.swPointerTargetProps.functionPointerSignature shall refer to BswModuleEntrys which each resolve to the

same function signature `at the time when the contract phase generation is executed`.

⌋*()*

[constr_1059] Compatibility of data types with `category VALUE` ⌈An `ApplicationDataType` of `category VALUE` shall (after all indirections created by `ImplementationDataType`s of `category TYPE_REFERENCE` are resolved) only be mapped/connected to an `ImplementationDataType` which also has `category VALUE`.

This rule shall be imposed **`at the time when the contract phase generation is executed`**.

⌋*()*

[constr_1060] Compatibility of data types with `category ARRAY, VAL_BLK` ⌈

`ApplicationDataType`	`ImplementationDataType`: **Array of uint8**	`ImplementationDataType`: **Array of other**
`ApplicationArrayDataType` of `category ARRAY, VAL_BLK`, `arraySizeSemantics` = `fixedSize`	`ImplementationDataType` of `category ARRAY`, with `ImplementationDataTypeElement` with `arraySizeSemantics` = `fixedSize`	`ImplementationDataType` of `category ARRAY`, with `ImplementationDataTypeElement` with `arraySizeSemantics` = `fixedSize`
`ApplicationArrayDataType` of `category ARRAY, VAL_BLK`, `arraySizeSemantics` = `variableSize`	`ImplementationDataType` of `category ARRAY`, with `ImplementationDataTypeElement` with `arraySizeSemantics` = `variableSize` or Variable-Size Array Data Type	Variable-Size Array Data Type

⌋*()*

[constr_1061] Compatibility of data types with `category STRUCTURE` ⌈An `ApplicationDataType` of `category STRUCTURE` shall (after all indirections created by `ImplementationDataType`s of `category TYPE_REFERENCE` are resolved) only be mapped/connected to an `ImplementationDataType` of `category STRUCTURE`.

This rule shall be imposed **`at the time when the contract phase generation is executed`**.

⌋*()*

[constr_1063] Compatibility of data types with `category BOOLEAN` ⌈An `ApplicationDataType` of `category BOOLEAN` shall (after all indirections created by `ImplementationDataType`s of `category TYPE_REFERENCE` are resolved) only be mapped/connected to an `ImplementationDataType` of `category VALUE`.

This rule shall be imposed **`at the time when the contract phase generation is executed`**.

⌋*()*

[constr_1064] Compatibility of data types with `category COM_AXIS`, `RES_AXIS`, `CURVE`, `MAP`, `CUBOID`, `CUBE_4`, or `CUBE_5` ⌈An `ApplicationDataType` of `category`

- `COM_AXIS`,
- `RES_AXIS`,
- `CURVE`,
- `MAP`,
- `CUBOID`,
- `CUBE_4`, or
- `CUBE_5`

shall (after all indirections created by `ImplementationDataType`s of `category TYPE_REFERENCE` are resolved) only be mapped/connected to an `ImplementationDataType` of `category`

- `STRUCTURE` or
- `ARRAY`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1066] Forbidden mappings to `ImplementationDataType` ⌈An `ApplicationDataType` shall never be mapped to

- an `ImplementationDataType` of `category`
  - `UNION`,
  - `DATA_REFERENCE`, or
  - `FUNCTION_REFERENCE`,
- or to an `ImplementationDataType` that contains `subElement`s of `category`
  - `UNION`,
  - `DATA_REFERENCE`, or
  - `FUNCTION_REFERENCE`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1068] Compatibility of `VariableDataPrototype`s or `ParameterDataPrototype`s typed by primitive data types ⌈Two `VariableDataPrototype`s or

`ParameterDataPrototype`s of `ApplicationPrimitiveDataType`s or `ImplementationDataType`s of `category` `VALUE`, `BOOLEAN`, or `STRING` are compatible **at the time when the RTE is generated** if and only if one of the following conditions applies:

1. All the following subconditions apply:

   (a) They are typed by (read "refer to") compatible `AutosarDataType`s

   (b) The two `VariableDataPrototype`s or `ParameterDataPrototype`s have identical `shortName`s. This is required to map `VariableDataPrototype`s in unordered `SenderReceiverInterface`s, `NvDataInterface`s and `ParameterInterface`s.

   (c) The attribute `swImplPolicy` is either set to `queued` for both or none of the `VariableDataPrototype`s.

2. In the context of a `DataPrototypeMapping`, one of the applicable `VariableDataPrototype`s or `ParameterDataPrototype`s is referenced by the `DataPrototypeMapping` in the role `firstDataPrototype` and the other `VariableDataPrototype`s or `ParameterDataPrototype`s is referenced by the same `DataPrototypeMapping` in the role `secondDataPrototype`.

⌋*()*

[constr_1069] Compatibility of `PortPrototype`s of different `DataInterface`s in the context of `AssemblySwConnector`s ⌈`PortPrototype`s of different `DataInterface`s are compatible **at the time when the RTE is generated** if and only if

1. One of the following conditions applies:

   (a) For each `VariableDataPrototype` or `ParameterDataPrototype` defined in the context of the `DataInterface` of the required `PortPrototype` a compatible (see [constr_1068]) `VariableDataPrototype` or `ParameterDataPrototype` exists in the `DataInterface` of the provided `PortPrototype`.

   The `shortName`s of `VariableDataPrototype`s and `ParameterDataPrototype`s are used to identify the pair.

   (b) A `VariableAndParameterInterfaceMapping`.`dataMapping` exists for which the following conditions apply:

   i. It is referenced by the corresponding `SwConnector`.

   ii. It references one of the two `VariableDataPrototype`s or `ParameterDataPrototype`s in the role `firstDataPrototype` and the other in the role `secondDataPrototype`.

2. For each such pair, the values of their `isService` attributes are identical.

⌋*()*

[constr_1070] Compatibility of PortPrototypes of different DataInterfaces in the context of DelegationSwConnectors ⌈PortPrototypes of different DataInterfaces are compatible **at the time when the RTE is generated** if and only if

1. One of the following conditions applies:

   (a) For each VariableDataPrototype or ParameterDataPrototype defined in the context of the DataInterface of the required inner Port-Prototype a compatible VariableDataPrototype or ParameterDat-aPrototype exists in the DataInterface of the required outer Port-Prototype.

   The shortName of VariableDataPrototypes and ParameterDat-aPrototypes are used to identify the pair.

   [constr_1071] defines which PortInterface elements are compatible depending on the PortInterface type and the swImplPolicy attributes of the PortInterface elements.

   (b) A VariableAndParameterInterfaceMapping.dataMapping exists for which the following conditions apply:

      i. It is referenced by the corresponding SwConnector.

      ii. It references one of the two VariableDataPrototypes or Param-eterDataPrototypes in the role firstDataPrototype and the other in the role secondDataPrototype.

2. One of the following conditions applies:

   (a) For at least one VariableDataPrototype or ParameterDataProto-type defined in the context of the SenderReceiverInterface, Nv-DataInterface or ParameterInterface of the provided inner Port-Prototype a compatible VariableDataPrototype or ParameterDat-aPrototype exists in the SenderReceiverInterface, NvDataInter-face or ParameterInterface of the provided outer PortPrototype.

   The shortNames of VariableDataPrototypes and ParameterDat-aPrototypes are used to identify the pair.

   [constr_1071] defines which PortInterface elements are compatible depending on the PortInterface type and the swImplPolicy attributes of the PortInterface elements.

   (b) A VariableAndParameterInterfaceMapping.dataMapping exists for which the following conditions apply:

      i. It is (if a corresponding SwConnector already exists) referenced by the corresponding SwConnector.

      ii. It references one of the two `VariableDataPrototype`s or `Param-eterDataPrototype`s in the role `firstDataPrototype` and the other in the role `secondDataPrototype`.

3. For each such pair, the values of their `isService` attributes are identical.

⌋*()*

[constr_1071] compatibility of `ParameterDataPrototype` and `VariableDat-aPrototype` ⌈

Provided Port Required Outer Port Provided Inner Port Required Outer Port			Required Port / Required Inner Port / Provided Outer Port / Provided Outer Port				
`PortInterface`			Prm			S/R	NvD
Interface Element			PDP			VDP	VDP
		`SwImplPolicyEnum`	`fixed`	`const`	`standard`	`standard`	`queued`	`standard`
Prm	PDP	`fixed`	yes	yes	yes	yes	no	yes
		`const`	no	yes	yes	yes	no	yes
		`standard`	no	no	yes	yes	no	yes
S/R	VDP	`standard`	no	no	no	yes	no	yes
		`queued`	no	no	no	no	yes	no
NvD	VDP	`standard`	no	no	no	yes	no	yes

⌋*()*

[constr_1072] Compatibility of `ModeSwitchInterface`s in the context of an `AssemblySwConnector` ⌈ `PortPrototype`s of different `ModeSwitchInterface`s are compatible **at the time when the RTE is generated** if and only if

1. One of the following conditions applies:

    (a) For the `ModeDeclarationGroupPrototype` defined in the context of the `ModeSwitchInterface` of the required `PortPrototype` a compatible `ModeDeclarationGroupPrototype` exists in the `ModeSwitchInter-face` of the provided `PortPrototype`.

    (b) A `ModeInterfaceMapping`.`modeMapping` exists for which the following conditions apply:

        i. It is referenced by the corresponding `SwConnector`.

        ii. It references one of the two `ModeDeclarationGroupPrototype`s in the role `firstModeGroup` and the other in the role `secondMod-eGroup`.

2. For each such pair, the values of their `isService` attributes are identical.

⌋*()*

[constr_1073] Compatibility of `ModeSwitchInterface`s in the context of an `DelegationSwConnector` ⌈ `PortPrototype`s of different `ModeSwitchInter-`

`face`s are compatible **at the time when the RTE is generated** if and only if

1. One of the following conditions applies:

   (a) For the `ModeDeclarationGroupPrototype` defined in the context of the `ModeSwitchInterface` of the inner `PortPrototype` a compatible `ModeDeclarationGroupPrototype` exists in the `ModeSwitchInterface` of the outer `PortPrototype`.

   (b) A `ModeInterfaceMapping.modeMapping` exists for which the following conditions apply:

      i. It is referenced by the corresponding `SwConnector`.

      ii. It references one of the two `ModeDeclarationGroupPrototype`s in the role `firstModeGroup` and the other in the role `secondModeGroup`.

2. For each such pair, the values of their `isService` attributes are identical.

⌋*()*

[constr_1074] Compatibility of `ModeDeclarationGroupPrototype`s ⌈`ModeDeclarationGroupPrototype`s are compatible **at the time when the RTE is generated** if and only if one of the following conditions applies:

1. They are typed by (read "refer to") compatible `ModeDeclarationGroup`s.

2. A `ModeDeclarationGroupPrototypeMapping` exists that identifies the differently named `ModeDeclarationGroupPrototype`s that correlate with each other. [constr_1210] applies.

⌋*()*

[constr_1075] Compatibility of `ModeDeclarationGroup`s ⌈`ModeDeclarationGroup`s are compatible **at the time when the RTE is generated** if and only if one of the following conditions applies:

1. All the following subconditions apply:

   (a) They define an identical number of `ModeDeclaration`s.

   (b) Each `ModeDeclaration` on the required side corresponds to a `ModeDeclaration` on the provided side with an identical `shortName`.

   (c) The `initialMode`s on both sides refer to `ModeDeclaration`s with identical `shortName`s.

   (d) The attribute `ModeDeclarationGroup.modeUserErrorBehavior.errorReactionPolicy` has identical values on both sides.

(e) The attribute `ModeDeclarationGroup.modeManagerErrorBehavior.errorReactionPolicy` has identical values on both sides.

(f) The attribute `ModeDeclarationGroup.modeUserErrorBehavior.defaultMode` either does not exist on both sides or refers on both sides to `ModeDeclaration`s with identical `shortName`s.

(g) The attribute `ModeDeclarationGroup.modeManagerErrorBehavior.defaultMode` either does not exist on both sides or refers on both sides to `ModeDeclaration`s with identical `shortName`s.

(h) one of the following subconditions applies:

- the attribute `category` has the value `ALPHABETIC_ORDER` on both sides.

- the attribute `category` has the value `EXPLICIT_ORDER` on both sides **and** the matching `ModeDeclaration`s according to 1(b) have the identical values of the attributes `ModeDeclaration.value` **and** also the value of `ModeDeclarationGroup.onTransitionValue` matches on both sides.

2. A `ModeDeclarationMapping` is applied which identifies the corresponding `ModeDeclaration`s.

In addition, the compatibility of corresponding `ModeTransition`s shall be checked, i.e. [constr_1194] and [constr_1245] apply.

⌋*()*

[constr_1076] Compatibility of `ArgumentDataPrototype`s ⌈Two `ArgumentDataPrototype`s are compatible **at the time when the RTE is generated** if and only if

1. They are typed by compatible `AutosarDataType`s **or** a `ClientServerOperationMapping.argumentMapping` exists that references one `ArgumentDataPrototype` in the role `firstDataPrototype` and the other `ArgumentDataPrototype` in the role `secondDataPrototype`.

2. They have the same value of the argument `direction` (`in`, `out` or `inout`), i.e. [constr_1268] applies.

⌋*()*

[constr_1077] Compatibility of `ApplicationError`s ⌈Two `ApplicationError`s are compatible **at the time when the RTE is generated** if and only if one of the following conditions applies:

1. All the following subconditions apply:

(a) They have the same `shortName`.

(b) They have the same attributes. Especially the `errorCode` shall be identical in both `ApplicationError`s.

2. A `ClientServerInterfaceMapping.errorMapping` exists that references one of the `ApplicationError`s in the role `firstApplicationError` and the other `ApplicationError`s in the role `secondApplicationError`.

⌋*()*

[constr_1078] Compatibility of `ClientServerOperation`s ⌈Two `ClientServerOperation`s are considered compatible **at the time when the RTE is generated** if their signatures match. In particular, they are compatible if and only if

1. They have the same number of `ArgumentDataPrototype`s.

2. The n-th arguments of both `ClientServerOperation`s are compatible. This implies ordering of `ArgumentDataPrototype`s.

3. They have identical values of attribute `diagArgIntegrity` or the attribute `diagArgIntegrity` does not exist on both sides.

4. They have the same `shortName` (again allows for mapping in `PortInterface`s).

5. The required `ClientServerOperation` specifies a compatible `ApplicationError` for each `ApplicationError` that is possibly raised by the provided `ClientServerOperation`, maybe more. Thereby, `ClientServerOperation`s that refer to a `possibleError` that represents the value `E_OK` are compatible to `ClientServerOperation`s that do refer to `possibleError`s where none of them represents the value `E_OK`.

⌋*()*

[constr_1079] Compatibility of `ClientServerInterface`s in the context of an `AssemblySwConnector` ⌈`ClientServerInterface`s are compatible **at the time when the RTE is generated** if and only if

1. One of the following conditions applies:

   (a) For each `ClientServerOperation` defined in the context of the `ClientServerInterface` of the required `PortPrototype` a compatible `ClientServerOperation` exists in the `ClientServerInterface` of the provided `PortPrototype`. The `shortName`s of `ClientServerOperation`s are used to identify the pair.

   (b) A `ClientServerInterfaceMapping.operationMapping` exists for which the following conditions apply:

      i. It is referenced by the corresponding `SwConnector`.

      ii. It references one of the two `ClientServerOperation`s in the role `firstOperation` and the other in the role `secondOperation`.

2. For each such pair, the values of their `isService` attributes are identical.

⌋*()*

[constr_1080] Compatibility of `ClientServerInterface`s in the context of an `DelegationSwConnector` ⌈`ClientServerInterface`s are compatible **at the time when the RTE is generated** if and only if

1. One of the following conditions applies:

   (a) For each `ClientServerOperation` defined in the context of the `ClientServerInterface` of the required inner `PortPrototype` a compatible `ClientServerOperation` exists in the `ClientServerInterface` of the required outer `PortPrototype`. The `shortName`s of `ClientServerOperation`s are used to identify the pair.

   (b) A `ClientServerInterfaceMapping`.`operationMapping` exists for which the following conditions apply:

      i. It is referenced by the corresponding `SwConnector`.

      ii. It references one of the two `ClientServerOperation`s in the role `firstOperation` and the other in the role `secondOperation`.

2. One of the following conditions applies:

   (a) For at least one `ClientServerOperation` defined in the context of the `ClientServerInterface` of the provided inner `PortPrototype` a compatible `ClientServerOperation` exists in the `ClientServerInterface` of the provided outer `PortPrototype`. The `shortName`s of `ClientServerOperation`s are used to identify the pair.

   (b) A `ClientServerInterfaceMapping`.`operationMapping` exists for which the following conditions apply:

      i. It is referenced by the corresponding `SwConnector`.

      ii. It references one of the two `ClientServerOperation`s in the role `firstOperation` and the other in the role `secondOperation`.

3. For each such pair, the values of their `isService` attributes are identical.

⌋*()*

[constr_1081] Compatibility of `TriggerInterface`s in the context of an `AssemblySwConnector` ⌈`TriggerInterface`s are compatible **at the time when the RTE is generated** if and only if

1. One of the following conditions applies:

   (a) For each `Trigger` defined in the context of the `TriggerInterface` of the required `PortPrototype` a compatible `Trigger` exists in the `TriggerInterface` of the provided `PortPrototype`. The `shortName`s of `Trigger` are used to identify the pair.

(b) A `TriggerInterfaceMapping.triggerMapping` exists for which the following conditions apply:

    i. It is referenced by the corresponding `SwConnector`.

    ii. It references one of the two `Trigger`s in the role `firstTrigger` and the other in the role `secondTrigger`.

2. For each such pair, the values of their `isService` attributes are identical.

⌋*()*

[constr_1082] Compatibility of `TriggerInterface`s in the context of an `DelegationSwConnector` ⌈`TriggerInterface`s are compatible **at the time when the RTE is generated** if and only if all the following conditions apply:

1. One of the following subconditions applies:

(a) For each `Trigger` defined in the context of the `TriggerInterface` of the **required** inner `PortPrototype` a compatible `Trigger` exists in the `TriggerInterface` of the **required** outer `PortPrototype`. The `shortName`s of `Trigger` are used to identify the pair.

(b) For at least one `Trigger` defined in the context of the `TriggerInterface` of the **provided** outer `PortPrototype` a compatible `Trigger` exists in the `TriggerInterface` of the **provided** inner `PortPrototype`. The `shortName`s of `Trigger` are used to identify the pair.

(c) A `TriggerInterfaceMapping.triggerMapping` exists for which all the following conditions apply:

    i. It is referenced by the corresponding `SwConnector`.

    ii. It references one of the two `Trigger`s in the role `firstTrigger` and the other in the role `secondTrigger`.

2. For each such pair, the values of their `isService` attributes are identical.

⌋*()*

[constr_1083] Compatibility of `Trigger`s ⌈`Trigger`s are compatible **at the time when the RTE is generated** if one of the following conditions is fulfilled:

- They have an identical `shortName`.

- A `TriggerMapping` exists that references one of the `Trigger`s in the role `firstTrigger` and the other `Trigger` in the role `secondTrigger`.

⌋*()*

[constr_1084] delegation of a provided outer `PortPrototype` ⌈The delegation of a provided outer `PortPrototype` is properly defined if the following criteria are fulfilled:

1. For each `VariableDataPrototype` or `ParameterDataPrototype` present in the `SenderReceiverInterface`, `NvDataInterface`, or `Parameter-`

`Interface` of the provided outer `PortPrototype` at least one connection via `DelegationSwConnector` to a provided inner `PortPrototype` **or** `PassThroughSwConnector` to a required outer `PortPrototype` with a compatible `VariableDataPrototype` or `ParameterDataPrototype` in the `SenderReceiverInterface NvDataInterface` or `ParameterInterface` of the provided inner `PortPrototype` **or** required outer `PortPrototype` exists.

Either the `shortName`s of `VariableDataPrototype`s or `ParameterDataPrototype`s are used to identify the pair or a `PortInterfaceMapping` defines which differently named `PortInterface` elements correlate with each other.

2. For each `VariableDataPrototype` provided by a `PRPortPrototype` that is typed by a `SenderReceiverInterface` or `NvDataInterface` and that is referenced in the role `outerPort` by a `DelegationSwConnector` a corresponding `VariableDataPrototype` owned by an `innerPort` shall be provided by either a `PPortPrototype` or a `PRPortPrototype`.

Either the `shortName`s of `VariableDataPrototype`s are used to identify the pair or a `PortInterfaceMapping` defines which differently named `PortInterface` elements correlate with each other.

3. For the `ModeDeclarationGroupPrototype` present in the `ModeSwitchInterface` of the provided outer `PortPrototype` exactly one connection via `DelegationSwConnector` to a provided inner `PortPrototype` **or** `PassThroughSwConnector` to a required outer `PortPrototype` with a compatible `ModeDeclarationGroupPrototype` in the `ModeSwitchInterface` of the provided inner `PortPrototype` **or** required outer `PortPrototype` exists.

Either the `shortName`s of `ModeDeclarationGroupPrototype`s are used to identify the pair or a `PortInterfaceMapping` defines which differently named `PortInterface` elements correlate with each other.

4. For each `ClientServerOperation` present in the `ClientServerInterface` of the provided outer `PortPrototype` exactly one connection via `DelegationSwConnector` to a provided inner `PortPrototype` **or** `PassThroughSwConnector` to a required outer `PortPrototype` with a compatible `ClientServerOperation` in the `ClientServerInterface` of the provided inner `PortPrototype` **or** required outer `PortPrototype` exists.

Either the `shortName`s of `ClientServerOperation`s are used to identify the pair or a `PortInterfaceMapping` defines which differently named `PortInterface` elements correlate with each other.

5. For each `Trigger` present in the `TriggerInterface` of the provided outer `PortPrototype` exactly one connection via `DelegationSwConnector` to a provided inner `PortPrototype` **or** `PassThroughSwConnector` to a required

outer `PortPrototype` with a compatible `Trigger` in the `TriggerInterface` of the provided inner `PortPrototype` **or** required outer `PortPrototype` exists.

Either the `shortName`s of `Trigger`s are used to identify the pair or a `PortInterfaceMapping` defines which differently named `PortInterface` elements correlate with each other.

This constraint is associated with two possible imposition times:

- **at the time when the RTE is generated**

- **at the time when the creation of the CompositionSwComponentType is finished**

⌋*()*

[constr_1085] Compatibility in the case of a flat ECU extract ⌈`PortPrototype`s of different `SenderReceiverInterface`s, `NvDataInterface`s, and `ParameterInterface`s are compatible if and only if for at least one `VariableDataPrototype` or `ParameterDataPrototype` defined in the context of the `SenderReceiverInterface`, `NvDataInterface`, or `ParameterInterface` of the `RPortPrototype` a compatible `VariableDataPrototype` or `ParameterDataPrototype` exists in the `SenderReceiverInterface`, `NvDataInterface`, or `ParameterInterface` of the provided `PortPrototype`.

The compatibility of `PortInterface` elements depends on the kind of `PortInterface` and the `swImplPolicy` attributes of the `PortInterface` elements.

Either the `shortName`s of `VariableDataPrototype`s and `ParameterDataPrototype`s are used to identify the pair or a `PortInterfaceMapping` defines which differently named `PortInterface` elements correlate with each other.

⌋*()*

[constr_1086] `SwConnector` between two specific `PortPrototype`s ⌈Each pair of `PortPrototype`s can only be connected by one and only one `SwConnector` **at the time when the RTE is generated**.

⌋*()*

[constr_1087] `AssemblySwConnector` inside `CompositionSwComponentType` ⌈An `AssemblySwConnector` can only connect `PortPrototype`s of `SwComponentPrototype`s that are owned by the same `CompositionSwComponentType` **at any time in the workflow**.

⌋*()*

[constr_1088] `DelegationSwConnector` inside `CompositionSwComponentType` ⌈A `DelegationSwConnector` can only connect a `PortPrototype` of a `SwComponentPrototype` that is owned by the same `CompositionSwComponent-`

`Type` that also owns the connected delegation `PortPrototype` **at any time in the workflow**.

⌋*()*

[constr_1091] `RTEEvent`s that may reference a `WaitPoint` ⌈A `WaitPoint` shall only be referenced from the listed `RTEEvent`s:

- `DataReceivedEvent`

- `DataSendCompletedEvent`

- `ModeSwitchedAckEvent`

- `AsynchronousServerCallReturnsEvent`

This rule shall be imposed **at the time when the contract phase is executed**.

⌋*()*

[constr_1092] Restrictions for the `ParameterSwComponentType` ⌈The following restrictions exist for a `ParameterSwComponentType`:

- it shall never aggregate a `SwcInternalBehavior` and

- the only aggregated `PortPrototype`s shall be `PPortPrototype`s of type `ParameterInterface`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1093] Definition of textual strings ⌈An `ApplicationPrimitiveDataType` of `category` `STRING` shall have a `swTextProps` which determines the `arraySizeSemantics` and `swMaxTextSize`.

This rule shall be imposed **at the time when the contract phase generation is executed**

⌋*()*

[constr_1095] Values of `nDataSets` vs. `reliability` ⌈If the value of `nDataSets` is greater than 0, the value of `reliability` shall not be set to `errorCorrection` **at the time when the RTE is generated**.

⌋*()*

[constr_1096] `SwcModeSwitchEvent` and `WaitPoint` ⌈A `RunnableEntity` that has a `WaitPoint` shall not be referenced by a `SwcModeSwitchEvent` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1097] RunnableEntity that has a WaitPoint ⌈A RunnableEntity that has a WaitPoint shall not be referenced by an RTEEvent that has a reference in the role disabledMode **at the time when the RTE is generated**.

⌋*()*

[constr_1098] Mode switch and mode disabling ⌈A SwcModeSwitchEvent shall not simultaneously reference to the same ModeDeclaration in both the roles mode and disabledMode **at the time when the RTE is generated**.

⌋*()*

[constr_1100] Unconnected RPortPrototype typed by a DataInterface ⌈For any element in an unconnected RPortPrototype typed by a DataInterface, there shall be a requiredComSpec that defines an initValue **at the time when the RTE is generated**.

⌋*()*

[constr_1101] Mode-related communication ⌈An RPortPrototype typed by ModeSwitchInterface shall not be referenced by more than one SwConnector **at the time when the RTE is generated**.

⌋*()*

[constr_1102] ApplicationError in the scope of one SwComponentType ⌈If a SwComponentType has PortPrototypes typed by different ClientServerInterfaces with equal shortName and ApplicationErrors defined then the following condition applies: ApplicationErrors with the same shortName shall have **identical values** of errorCodes.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1103] NonqueuedReceiverComSpec and enableUpdate ⌈A NonqueuedReceiverComSpec that has the value of attribute enableUpdate set to true **at the time when the contract phase generation is executed** may not reference a dataElement that in turn is referenced by a VariableAccess in the role dataReadAccess.

⌋*()*

[constr_1104] Trigger communication shall not implement an n:1 pattern ⌈An RPortPrototype typed by a TriggerInterface shall not be connected to PortPrototypes typed by TriggerInterfaces such that a given Trigger in the TriggerInterface of the RPortPrototype is connected to more than one compatible (see [constr_1081], [constr_1082], and [constr_1251]) Trigger in the TriggerIn-

`terface`s of the connected `PortPrototype`s **at the time when the RTE is generated**.

⌋*()*

[constr_1105] Value of `arraySize` ⌈The value of the attribute `arraySize` of an `ImplementationDataTypeElement` owned by an `ImplementationDataType` or `ImplementationDataTypeElement` of `category` `ARRAY` shall be greater than 0 unless attribute `ImplementationDataTypeElement.arraySizeHandling` exists and is set to the value `inheritedFromArrayElementTypeSize`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1106] Structure shall have at least one element ⌈An `ImplementationDataType` or `ImplementationDataTypeElement` of `category` `STRUCTURE` shall own at least one `ImplementationDataTypeElement` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1107] Union shall have at least one element ⌈An `ImplementationDataType` or `ImplementationDataTypeElement` of `category` `UNION` shall own at least one `ImplementationDataTypeElement` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1108] Existence and value of attribute `ApplicationError.errorCode` ⌈Attribute `ApplicationError.errorCode` shall exist **at the time when the contract phase generation is executed** and its value shall not exceed the closed interval 1 .. 63.

The following exception applies: **only** if attribute `possibleError` is supposed to represent the error code `E_OK`, the value 0 shall be allowed.

⌋*()*

[constr_1109] Mapping of `SwComponentPrototype`s typed by a `SensorActuatorSwComponentType` ⌈A `SwComponentPrototype` typed by a `SensorActuatorSwComponentType` needs to be mapped and run on exactly that ECU that contains the `HwElement` corresponding to the `HwType` that its `SensorActuatorSwComponentType` refers to in case it accesses the hardware via the I/O hardware abstraction layer.

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_1111] Constraints of `dataId` in `PROFILE_01` ⌈In `PROFILE_01`, there shall be only one element in the set and the applicable range of values is [0 .. 65535].

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1112] Constraints of `dataIdMode` in `PROFILE_01` ⌈In `PROFILE_01`, the applicable range of values for `dataIdMode` is [0 .. 3].

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1113] Existence of attributes in `PROFILE_01` ⌈In `PROFILE_01`, the following attributes shall exist:

- `dataLength`

- `dataId`

at the time when the contract phase generation is executed.

⌋*()*

[constr_1114] Constraints of `crcOffset` in `PROFILE_01` ⌈In `PROFILE_01`, the applicable range of values for `crcOffset` is [0 .. 65535]. For the value of this attribute the constraint *value mod 4 = 0* applies.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1115] Constraints of `counterOffset` in `PROFILE_01` ⌈In `PROFILE_01`, the applicable range of values for `counterOffset` is [0 .. 65535]. For the value of this attribute the constraint *value mod 4 = 0* applies.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1116] Constraints of `dataLength` in `PROFILE_01` ⌈In `PROFILE_01`, the applicable range of values for `dataLength` is [0 .. 240]. For the value of this attribute the constraint *value mod 8 = 0* applies.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1117] Constraints of `maxDeltaCounterInit` in `PROFILE_01` ⌈In `PROFILE_01`, the applicable range of values for `EndToEndDescription.maxDeltaCounterInit` and `ReceiverComSpec.maxDeltaCounterInit` is [0 .. 14].

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1118] Existence of attributes in `PROFILE_02` ⌈In `PROFILE_02`, only the following attributes shall exist:

- `dataLength`
- `dataId`

at the time when the contract phase generation is executed

⌋*()*

[constr_1119] Constraints of `dataLength` in `PROFILE_02` ⌈In `PROFILE_02`, the applicable range of values for `dataLength` is [0 .. 65535]. For the value of this attribute the constraint *value mod 8 = 0* applies.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1120] Constraints of `dataId` in `PROFILE_02` ⌈In `PROFILE_02`, there shall be exactly ordered 16 elements in the set and the applicable range of values is [0 .. 255].

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1121] Constraints of `maxDeltaCounterInit` in `PROFILE_02` ⌈In `PROFILE_02`, the applicable range of values for `EndToEndDescription.maxDeltaCounterInit` and `ReceiverComSpec.maxDeltaCounterInit` is [0 .. 15].

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1126] Compatibility of `DataConstr`s ⌈The `DataConstr` (e.g. the limits) defined by the type of the providing data element shall be within the constraints defined by the type of the requiring data element.

For client-server communication, the following rules apply:

- For `argument`s with attribute `direction` set to the value `in`, the client shall take the role of the *provider* and the server shall take the role of the *requiring side*.

- For `argument`s with attribute `direction` set to the value `inout` the `DataConstr` shall be equal on both sides.

- For `argument`s with attribute `direction` set to the value `out`, the server shall take the role of the *provider* and the client shall take the role of the *requiring side*.

This rule shall be applied **at the time when the RTE is generated**.

⌋*()*

[constr_1128] Queue length of `ClientServerOperation`s associated with the same `RunnableEntity` ⌈If two or more `OperationInvokedEvent`s reference a single `RunnableEntity` the value of the `ServerComSpec` attribute `queueLength` shall be **identical** for all `ServerComSpec`s owned by `PPortPrototype`s of the enclosing `SwComponentType` that reference one of the `ClientServerOperation`s that are also referenced by the `OperationInvokedEvent`s.

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_1129] `swImplPolicy` and `NonqueuedReceiverComSpec` ⌈The attribute `swImplPolicy` of a `dataElement` referenced by a `NonqueuedReceiverComSpec` **shall not** be set to the value `queued` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1130] `swImplPolicy` and `QueuedReceiverComSpec` ⌈The attribute `swImplPolicy` of a `dataElement` referenced by a `QueuedReceiverComSpec` **shall** be set to the value `queued` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1131] `swImplPolicy` and `NonqueuedSenderComSpec` ⌈The attribute `swImplPolicy` of a `dataElement` referenced by a `NonqueuedSenderComSpec` **shall not** be set to the value `queued` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1132] `swImplPolicy` and `QueuedSenderComSpec` ⌈The attribute `swImplPolicy` of a `dataElement` referenced by a `QueuedSenderComSpec` **shall** be set to the value `queued` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1134] Allowed structure of `TEXTTABLE` ⌈The existence of `physConstrs` is not allowed and `compuInternalToPhys` shall exist with `compuScale`s consisting of `upperLimit` and `lowerLimit` **at any time in the workflow**.

⌋*()*

[constr_1135] Limit of `vt` in `BITFIELD_TEXTTABLE` ⌈The separator for splitting the string representing the value is "|" and is therefore forbidden to appear in `vt` **at any time in the workflow**.

⌋*()*

[constr_1137] Applicability of `ParameterInterface` ⌈A `PPortPrototype` typed by a `ParameterInterface` can **only** be owned by a `ParameterSwComponentType` or a `CompositionSwComponentType` **at any time in the workflow**.

⌋*()*

[constr_1138] `SwcServiceDependency.assignedPort` and `DiagEventDebounceMonitorInternal` ⌈If a `SwcServiceDependency` aggregates `DiagnosticEventNeeds` in the role `serviceNeeds`, then an `assignedPort` with attribute `role` set to the value `CallbackGetFaultDetectCounter` shall only exist if the monitor implements internal debouncing, i.e concrete subclass `DiagEventDebounceMonitorInternal` is aggregated in the role `DiagnosticEventNeeds`. `diagEventDebounceAlgorithm`.

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_1140] Combination of `invalidValue` with the attribute `handleInvalid` ⌈The combination of setting the attribute `handleInvalid` of the meta-class `InvalidationPolicy` owned by `SenderReceiverInterface` to value `replace` **and** of setting the value of the attribute `initValue` owned by a corresponding `NonqueuedReceiverComSpec` effectively to the value of the `invalidValue` (owned by a corresponding `SwDataDefProps`) is not supported **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1141] Applicability of the `scope` attribute ⌈The attribute `scope` of meta-class `VariableAccess` shall **only** be applied with respect to the aggregation of `VariableAccess` in the following roles:

- `dataReadAccess`

- `dataWriteAccess`

- `dataSendPoint`

- `dataReceivePointByValue`

- `dataReceivePointByArgument`

This rule shall be imposed **`at the time when the contract phase genera-`**
`tion is executed`.

⌋*()*

[constr_1142] `category` of `CompuMethod` shall not be extended ⌈In contrast to the
general rule that `category` can be extended by user-specific values it is **not allowed**
to extend the meaning of the attribute `category` of meta-class `CompuMethod` **`at any`**
`time in the workflow`.

⌋*()*

[constr_1144] `SensorActuatorSwComponentType`, `EcuAbstractionSwCompo-`
`nentType`, and `ComplexDeviceDriverSwComponentType` may only reference a
`HwType` ⌈The attribute `sensorActuator` of `SensorActuatorSwComponentType`,
the attribute `hardwareElement` of `EcuAbstractionSwComponentType`, and the
attribute `hardwareElement` of `ComplexDeviceDriverSwComponentType` may
only reference a `HwType`. References to other subclasses of `HwDescriptionEn-`
`tity` are not allowed **`at the time when the RTE is generated`**.

⌋*()*

[constr_1146] Applicability of a symbol for a `CompuScale` in C code ⌈The `sym-`
`bol` attribute shall only be provided for `CompuScale`s where the `category` of the
enclosing `CompuMethod` is one of the following:

- `TEXTTABLE`

- `SCALE_LINEAR_AND_TEXTTABLE`

- `SCALE_RATIONAL_AND_TEXTTABLE`

- `BITFIELD_TEXTTABLE`

This rule shall be imposed **`at the time when the contract phase genera-`**
`tion is executed`.

⌋*()*

[constr_1147] Standardized values for the attribute `category` of meta-class
`PortGroup` ⌈The following values of the attribute `category` of meta-class `Port-`
`Group` are reserved by the AUTOSAR standard:

- `MODE_MANAGEMENT`: This represents the usage of the `PortGroup` for the pur-
  pose of mode management

- `PARTIAL_NETWORKING`: This represents the usage of the `PortGroup` for the
  purpose of partial networking

This rule shall be imposed **`at the time when the contract phase genera-`**
`tion is executed`.

⌋*()*

[constr_1148] **PortInterfaces of PortPrototypes used to connect to NvBlockSwComponentTypes** ⌈PortInterfaces of PortPrototypes used to connect to NvBlockSwComponentTypes as well as the PortInterfaces used in the context of NvBlockSwComponentTypes shall **always** set the value of the attribute isService to false **at the time when the RTE is generated**.

⌋*()*

[constr_1149] **PortPrototypes used for NV data management** ⌈A PortPrototype typed by a ClientServerInterface used for NV data management, i.e. the interaction of ApplicationSwComponentTypes with NvBlockSwComponentTypes, shall be typed by ClientServerInterfaces that are compatible to the particular ClientServerInterfaces derived from MOD_GeneralBlueprints [7]. [constr_1148] applies.

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_1150] Usage of valueType for PortDefinedArgumentValue ⌈The valueType (typically this boils down to integer values used to specify an "id") associated with PortDefinedArgumentValue shall be of category VALUE or TYPE_REFERENCE. The latter case is only supported if the value of category of the target data type is set to VALUE.

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_1151] Applicability of PortInterfaceMapping ⌈A PortInterfaceMapping is only applicable and valid for a SwConnector if the two PortPrototypes which are referenced by the SwConnector are typed by the same two PortInterfaces which are mapped by the PortInterfaceMapping. This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_1152] category of ApplicationArrayElement and AutosarDataType referenced in the role type shall be kept in sync ⌈The value of category of an ApplicationArrayElement shall always be identical to the value of category of the AutosarDataType referenced by the ApplicationArrayElement.

⌋*()*

[constr_1153] Applicability of compatibility requirements for CompuScales ⌈Compatibility requirements for CompuScales shall only apply for CompuScales where the category of the enclosing CompuMethod is one of the following:

- TEXTTABLE

- SCALE_LINEAR_AND_TEXTTABLE

- SCALE_RATIONAL_AND_TEXTTABLE

- TAB_NOINTP

- BITFIELD_TEXTTABLE

- LINEAR

- RAT_FUNC

- IDENTICAL

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_1154] Compatibility of `CompuScale`s for sender-receiver communication and similar use cases ⌈For sender-receiver communication and similar use cases, it is required that the set of `CompuScale`s defined in the `CompuMethod` of the provider of the communication (i.e. on the side of the `PPortPrototype`) shall be a subset of the set of `CompuScale`s defined in the `CompuMethod` on the required side (i.e. on the side of the `RPortPrototype`) **at the time when the RTE is generated**.

⌋*()*

[constr_1155] Compatibility of `CompuScale`s for client-server communication ⌈For client-server communication, the following rules apply **at the time when the RTE is generated**:

For `argument`s of direction `IN` the `CompuScale`s defined in the `CompuMethod` of the client (i.e. on the side of the `RPortPrototype`) shall be a subset of the set of `CompuScale`s defined in the `CompuMethod` supported at the server (i.e. on the side of the `PPortPrototype`).

For `argument`s of the direction `OUT` the set of `CompuScale`s defined in the `CompuMethod` of the server (i.e. on the side of the `PPortPrototype`) shall be a subset of the set of `CompuScale`s defined in the `CompuMethod` supported at the client (i.e. on the side of the `RPortPrototype`).

For `argument`s of direction `INOUT` the set of `CompuScale`s defined in the `CompuMethod` of server and client shall be identical.

⌋*()*

[constr_1156] Relevance of "names" of `CompuScale`s ⌈`CompuScale`s which contribute to tabular conversion by having a `compuConst` are compatible **if and only if** the "names" of the `compuScale`s, (namely `shortLabel`, `vt` and `symbol`, according to the priority rules communicated in [TPS_SWCT_01431]) are equal.

If the scale has no `compuConst`, "names" of `CompuScale`s are not relevant for compatibility.

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_1158] Applicable `category`s for attribute `ImplementationDataType`. `swDataDefProps.compuMethod` ⌈

	IDENTICAL	LINEAR	SCALE_LINEAR	SCALE_LINEAR_AND_TEXTTABLE	RAT_FUNC	SCALE_RATIONAL_AND_TEXTTABLE	TEXTTABLE	TAB_NOINTP	BITFIELD_TEXTTABLE
VALUE	n/a	n/a	n/a	n/a	n/a	n/a	x	n/a	x
TYPE_REFERENCE	n/a	n/a	n/a	n/a	n/a	n/a	x	n/a	x

⌋*()*

[constr_1159] Consistency of `VariableAndParameterInterfaceMapping` with respect to the referenced `DataInterface`s ⌈Within one `VariableAndParameterInterfaceMapping` all `firstDataPrototype`s shall belong to one and only one `DataInterface` and all `secondDataPrototype`s shall belong to one other and only one other `DataInterface` **at any time in the workflow**.

⌋*()*

[constr_1161] Applicability of the attribute `Ref.index` ⌈The usage of attribute `Ref.index` is limited to references to the following meta-classes:

- `ApplicationArrayElement`
- Sub-classes of `AbstractImplementationDataTypeElement`.

This rule shall be imposed **at any time in the workflow**.

⌋*()*

[constr_1163] Compatibility of `CompuMethod`s ⌈Two `CompuMethod` definitions are compatible **at the time when the RTE is generated** if and only if all attributes **except**

- `shortName`
- `desc`
- `introduction`
- `longName`
- `adminData`
- `annotation`

- `displayFormat`

are **identical and** the `compuScale`s and `unit`s are compatible.

⌋*()*

[constr_1164] Number of `argument`s owned by a `RunnableEntity` ⌈If a given `RunnableEntity` owns `RunnableEntityArgument`s in the role `argument`, then the number of these `RunnableEntityArgument`s shall be identical to the number of applicable `portArgValue`s of the `PortAPIOption` that references the `PortPrototype` that in turn is referenced by the `OperationInvokedEvent` that references the `RunnableEntity` **plus** the number of `ArgumentDataPrototype`s aggregated in the role `argument` by the `ClientServerOperation` referenced by said `OperationInvokedEvent` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1165] Applicability of `RunnableEntityArgument` ⌈The existence of a `RunnableEntityArgument` is limited to `RunnableEntity`s triggered by a `ClientServerOperation`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1166] Restrictions of `ModeRequestTypeMap` ⌈For every `ModeDeclarationGroup` referenced by a `ModeDeclarationGroupPrototype` used in a `PortPrototype` typed by a `ModeSwitchInterface` a `ModeRequestTypeMap` shall exist that points to the `ModeDeclarationGroup` and also to an eligible `ImplementationDataType`.

The `ModeRequestTypeMap` shall be aggregated by a `DataTypeMappingSet` which is referenced from the `SwcInternalBehavior` that is owned by the `ApplicationSwComponentType` that also owns the `PortPrototype`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1167] `ImplementationDataType`s used as `ModeRequestTypeMap`.`implementationDataType` ⌈The `ImplementationDataType` referenced by a `ModeRequestTypeMap` shall **either** be

- of `category VALUE` **or**

- of `category TYPE_REFERENCE` that in turn references an `ImplementationDataType` of `category VALUE`.

The `baseType` referenced by the `ImplementationDataType` shall have set the value of the attribute `BaseTypeDirectDefinition`.`baseTypeEncoding` to `NONE`.

This rule shall be imposed **at the time when the contract phase genera-tion is executed**.

⌋*()*

[constr_1168] Compatibility of ImplementationDataTypes used in the Mod-eRequestTypeMap ⌈Both ImplementationDataTypes shall fulfill [constr_1167].

In addition to that, the possible numbers used for representing ModeDeclarations on the side of the mode manager shall match the supported range of the Imple-mentationDataType used for representing ModeDeclarations on the side of the mode user (see [constr_1075]).

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_1169] Allowed values for Trigger.swImplPolicy ⌈The **only** allowed val-ues for the attribute Trigger.swImplPolicy are either STANDARD (in which case the Trigger processing does not use a queue) or QUEUED (in which case the process-ing of Triggers positively uses a queue). **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1170] Existence of attribute EndToEndDescription.maxDeltaCoun-terInit for PROFILE_01 ⌈If the value of attribute EndToEndDescription.cat-egory is set to PROFILE_01 **and either**

- the condition described in [TPS_SWCT_01850] concerning the referenced VariableDataPrototype is not fulfilled **or**

- attribute RPortPrototype.requiredComSpec.maxDeltaCounterInit does not exist,

then attribute EndToEndProtection.endToEndProfile.maxDeltaCounterInit **shall exist at the time when the contract phase generation is exe-cuted**.

⌋*()*

[constr_1171] Existence of attribute EndToEndDescription.maxDeltaCoun-terInit for PROFILE_02 ⌈If the value of EndToEndDescription.category is set to PROFILE_02 **and either**

- the condition described in [TPS_SWCT_01850] concerning the referenced VariableDataPrototype is not fulfilled **or**

- attribute RPortPrototype.requiredComSpec.maxDeltaCounterInit does not exist,

then attribute EndToEndProtection.endToEndProfile.maxDeltaCounterInit **shall exist at the time when the contract phase generation is executed**.

⌋*()*

[constr_1172] Allowed values of SwCalibrationAccessEnum for ModeDeclarationGroupPrototype ⌈The only allowed values of swCalibrationAccess aggregated by ModeDeclarationGroupPrototype are

- notAccessible and

- readOnly.

This rule shall be imposed **at any time in the workflow**.

⌋*()*

[constr_1173] Applicability of AutosarParameterRef referencing a VariableDataPrototype ⌈A reference from AutosarParameterRef to VariableDataPrototype is **only** applicable if the AutosarParameterRef is used in the context of SwAxisGrouped **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1174] PortInterfaces used in the context of CompositionSwComponentTypes cannot refer to AUTOSAR services ⌈CompositionSwComponentTypes shall not own PortPrototypes typed by PortInterfaces where the attribute isService is set to true **at any time in the workflow**.

⌋*()*

[constr_1175] Depending on its category, CompuMethod shall refer to a unit ⌈As a CompuMethod specifies the conversion between the physical world and the numerical values, it shall refer to a unit unless the CompuMethod's category is one of TEXTTABLE, BITFIELD_TEXTTABLE, or IDENTICAL.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1176] Compatibility of CompuScales of category LINEAR and RAT_FUNC ⌈CompuScales of category LINEAR and RAT_FUNC are considered compatible **at the time when the RTE is generated** if they yield the same conversion.

⌋*()*

[constr_1177] Allowed targetCategory for SwPointerTargetProps ⌈If the value of attribute targetCategory exists, then it shall be set to one of the following values:

- TYPE_REFERENCE

- FUNCTION_REFERENCE

- VALUE (only applicable if the `SwPointerTargetProps.swDataDefProps` refers to a `SwBaseType` where attribute `nativeDeclaration` is set to the value "void")

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1178] Existence of attributes of `SwDataDefProps` in the context of `ImplementationDataType` ⌈For the sake of removing possible sources of ambiguity, `SwDataDefProps` used in the context of `ImplementationDataType` can **only have one of**

- baseType

- swPointerTargetProps

- implementationDataType

at the time when the contract phase generation is executed.

⌋*()*

[constr_1181] Numerical values used in `ModeDeclaration.value` and `ModeDeclarationGroup.onTransitionValue` ⌈The numerical values used to define the `value` attributes and the `onTransitionValue` attribute of a `ModeDeclarationGroup` shall not overlap **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1182] Allowed values for `InternalTriggeringPoint.swImplPolicy` ⌈The **only** allowed values for the attribute `swImplPolicy` of meta-class `InternalTriggeringPoint` are either STANDARD (in which case the processing of the internal triggering does not use a queue) or QUEUED (in which case the processing of internal triggering positively uses a queue).

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_1183]`EndToEndProtectionVariablePrototype`s aggregated by `EndToEndProtection` ⌈All `EndToEndProtectionVariablePrototype`s aggregated by the same `EndToEndProtection` shall refer to the identical `sender` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1184] Consistency of `rootDataPrototype` and `base` in the context of `ApplicationCompositeElementInPortInterfaceInstanceRef` ⌈The `rootDataPrototype` referenced by `ApplicationCompositeElementInPortInter-`

`faceInstanceRef` shall be owned by the applicable subclass of `DataInterface` referenced in the role `base`.

This implies that the `rootDataPrototype` shall be a `ParameterDataPrototype` if the `base` is a `ParameterInterface`. Otherwise, the `rootDataPrototype` shall be a `VariableDataPrototype`.

This rule shall be applied at **at any time in the workflow**.

⌋*()*

[constr_1185] Consistency of data types in the context of `ApplicationCompositeElementInPortInterfaceInstanceRef` ⌈The definition of attributes `contextDataPrototype` and `targetDataPrototype` shall (via the type-prototype pattern) be enclosed in the context of the definition of the data type used to type `rootDataPrototype` **at any time in the workflow**.

⌋*()*

[constr_1186] Consistency of data types in the context of `ArVariableInImplementationDataInstanceRef` ⌈The definition of attributes `contextDataPrototype` and `targetDataPrototype` shall be enclosed in the context of the definition of the data type used to type `rootVariableDataPrototype` **at any time in the workflow**.

⌋*()*

[constr_1187] Compatibility of `VariableDataPrototype`s or `ParameterDataPrototype`s typed by composite data types ⌈`DataPrototype`s of `ApplicationCompositeDataType`s or `ImplementationDataType`s of category `STRUCTURE` or `ARRAY` are compatible **at the time when the RTE is generated** if one of the following conditions evaluates to true:

1. The underlying `ApplicationCompositeDataType`s or `ImplementationDataType`s of category `STRUCTURE` or `ARRAY` are identical

2. The underlying `ApplicationCompositeDataType`s or `ImplementationDataType`s of category `STRUCTURE` or `ARRAY` fulfill the following condition:

   - They consist of the same number of elements **and**

   - They are composed of compatible `AutosarDataType`s (either `ApplicationCompositeDataType`s or `ImplementationDataType`s of category `STRUCTURE` or `ARRAY` **OR** `ApplicationPrimitiveDataType`s or `ImplementationDataType`s of category `VALUE`, `BOOLEAN`, or `STRING`) in *the same order* **and**

   - All attributes match exactly, except for the `shortName` of the M1 `AutosarDataType`.

3. In the context of a `DataPrototypeMapping`, for each `ApplicationCompositeElementDataPrototype` of the required `DataPrototype` a `SubEle-`

mentMapping exists such that a `ApplicationCompositeDataType-SubElementRef` in the role `firstElement` or `secondElement` exists that references the required `ApplicationCompositeElementDataPrototype` **and** a corresponding `ApplicationCompositeDataTypeSubElementRef` exists in the **other** role (i.e. `secondElement` or `firstElement`) that in turn references an `ApplicationCompositeElementDataPrototype` of the provided `ApplicationCompositeDataType`.

4. If and only if the `DataPrototype` is **not** typed by an `ApplicationDataType` but by an `ImplementationDataType`: in the context of a `DataPrototypeMapping`, for each `ImplementationDataTypeElement` of the required `DataPrototype` a `SubElementMapping` exists such that a `ImplementationDataTypeSubElementRef` in the role `firstElement` or `secondElement` exists that references the required `ImplementationDataTypeElement` **and** a corresponding `ImplementationDataTypeSubElementRef` exists in the **other** role (i.e. `secondElement` or `firstElement`) that in turn references an `ImplementationDataTypeElement` of the provided `ImplementationDataType`.

⌋*()*

[constr_1188] Existence of `ReceiverComSpec.replaceWith` ⌈The aggregation of `VariableAccess` in the role `ReceiverComSpec.replaceWith` shall exist **if and only if at least one of the following conditions is fulfilled**:

- Attribute `ReceiverComSpec.handleOutOfRange` is set to the value `externalReplacement`.

- Attribute `SenderReceiverInterface.invalidationPolicy.handleInvalid` is set to the value `externalReplacement`.

This rule shall be applied **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1190] Only one mapping for composite to primitive use case ⌈In the case described by [TPS_SWCT_01195] only one `subElementMapping` shall exist at the enclosing `DataPrototypeMapping` **at the time when the RTE is generated**.

⌋*()*

[constr_1191] Value of `Limit` shall yield a numerical value ⌈After all variability is bound, the content obtained from a limit shall yield a numerical value **at any time in the workflow**.

⌋*()*

[constr_1192] Compatibility of "`IDENTICAL`" to "`RAT_FUNC`" or "`LINEAR`" ⌈Similar to [constr_1176], a `CompuScale` where the `category` of the enclosing `CompuMethod`

is set to `IDENTICAL` is — **at the time when the RTE is generated** — considered compatible to a `CompuScale` where the `category` of the enclosing `CompuMethod` is set to `RAT_FUNC` or `LINEAR` if the following rule applies:

$$int = \frac{N_0 + N_1 * phys + N_i * phys^i}{D_0 + D_1 * phys + D_i * phys^i} = phys$$

⌋*()*

[constr_1193] `ModeDeclaration` shall be referenced by at least one `ModeTransition` in the role `enteredMode` ⌈For each `ModeDeclaration` at least one `ModeTransition` shall reference the `ModeDeclaration` in the role `enteredMode`.

This constraint shall apply **at the time when the RTE is generated only** if there is at least one `ModeTransition` defined in the context of the enclosing `ModeDeclarationGroup` and it shall **not** apply to the `initialMode`.

⌋*()*

[constr_1194] Identical `ModeTransition`s ⌈Two `ModeDeclarationGroup`s contain identical `modeTransition`s **at the time when the RTE is generated** if and only if

1. For each `ModeTransition` defined in the context of the mode provider one `ModeTransition` with the same `shortName` is defined in the context of the mode user.

2. Each pair of `ModeTransition`s in both `ModeDeclarationGroup`s identified by their respective `shortName` have identical targets (in terms of the `shortName` of the referenced `ModeDeclaration`) of the references `enteredMode` and `exitedMode`.

⌋*()*

[constr_1195] `SwcModeSwitchEvent` and the definition of `ModeTransition` ⌈For each pair of `ModeDeclaration`s referenced by a `SwcModeSwitchEvent` with attribute `activation` set to `onTransition` a `ModeTransition` shall be defined in the corresponding direction (i.e. from `exitedMode` to `enteredMode`). This constraint shall only apply **at the time when the RTE is generated** if the respective `ModeDeclarationGroup` defines at least one `modeTransition`.

⌋*()*

[constr_1196] Existence of `networkRepresentation` vs. `compositeNetworkRepresentation` ⌈If a `ReceiverComSpec` or `SenderComSpec` aggregates `networkRepresentation` it shall **not** aggregate `compositeNetworkRepresentation` (and vice versa) **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1197] Existence of `compositeNetworkRepresentation` shall be comprehensive ⌈If at least one `compositeNetworkRepresentation` exists then for

each leaf `ApplicationCompositeElementDataPrototype` of the affected `ApplicationCompositeDataType` exactly one `compositeNetworkRepresentation` shall be defined **at the time when the contract phase generation is executed**.

For each such `compositeNetworkRepresentation`, attributes `leafElement` and `networkRepresentation` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1200] Queued communication is not applicable for `dataElement`s owned by `PRPortPrototype` ⌈The `swImplPolicy` shall not be set to `queued` for any `dataElement` owned by a `PRPortPrototype` **at any time in the workflow**.

⌋*()*

[constr_1202] Supported connections by `AssemblySwConnector` between `PortPrototype`s typed by a `SenderReceiverInterface` or `NvDataInterface` ⌈

	RPortPrototype	PPortPrototype	PRPortPrototype
RPortPrototype	No	Yes	Yes
PPortPrototype	Yes	No	Yes
PRPortPrototype	Yes	Yes	Yes

⌋*()*

[constr_1203] Supported connections by `DelegationSwConnector` between `PortPrototype`s typed by a `SenderReceiverInterface` or `NvDataInterface` ⌈

innerPort	outerPort		
	RPortPrototype	PPortPrototype	PRPortPrototype
RPortPrototype	Yes	No	Yes
PPortPrototype	No	Yes	Yes
PRPortPrototype	Yes	Yes	Yes

⌋*()*

[constr_1204] Supported connections by `AssemblySwConnector` between `PortPrototype`s typed by a `ClientServerInterface`, `ModeSwitchInterface`, or `TriggerInterface` ⌈

	RPortPrototype	PPortPrototype	PRPortPrototype
RPortPrototype	No	Yes	Yes
PPortPrototype	Yes	No	No
PRPortPrototype	Yes	No	No

⌋*()*

[constr_1205] Supported connections by `DelegationSwConnector` between `PortPrototype`s typed by a `ClientServerInterface`, `ModeSwitchInterface`, or `TriggerInterface` ⌈

innerPort	outerPort		
	RPortPrototype	PPortPrototype	PRPortPrototype
RPortPrototype	Yes	No	No
PPortPrototype	No	Yes	No
PRPortPrototype	No	Yes	No

⌋*()*

[constr_1209] Mapping of `ModeDeclaration`s of mode user to `ModeDeclaration` of mode manager ⌈A configuration that maps **several** `ModeDeclaration`s representing modes of a mode user to **one** `ModeDeclaration` representing a mode of a mode manager shall be rejected **at any time in the workflow**.

⌋*()*

[constr_1210] Mapping of `ModeDeclaration`s of mode user to all `ModeDeclaration`s of mode manager ⌈If a `ModeDeclarationMapping` exists that references a `ModeDeclaration` representing a mode of the mode manager, then `ModeDeclarationMapping`s shall exist that map all modes of the mode manager to modes of the mode user **at the time when the RTE is generated**.

⌋*()*

[constr_1211] Constraints of `maxNoNewOrRepeatedData` in `PROFILE_01` ⌈In `PROFILE_01`, the applicable range of values for `EndToEndDescription.maxNoNewOrRepeatedData` and `ReceiverComSpec.maxNoNewOrRepeatedData` is [0 .. 14].

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1212] Constraints of `syncCounterInit` in `PROFILE_01` ⌈In `PROFILE_01`, the applicable range of values for `EndToEndDescription.syncCounterInit` and `ReceiverComSpec.syncCounterInit` is [0 .. 14].

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1213] Constraints of `maxNoNewOrRepeatedData` in `PROFILE_02` ⌈In `PROFILE_02`, the applicable range of values for `EndToEndDescription.maxNoNewOrRepeatedData` and `ReceiverComSpec.maxNoNewOrRepeatedData` is [0 .. 15].

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1214] Constraints of `syncCounterInit` in `PROFILE_02` ⌈In `PROFILE_-02`, the applicable range of values for `EndToEndDescription.syncCounterInit` and `ReceiverComSpec.syncCounterInit` is [0 .. 15].

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1215] Existence of attribute `EndToEndDescription.maxNoNewOrRepeatedData` for `PROFILE_01` ⌈If the value of attribute `EndToEndDescription.category` is set to `PROFILE_01` **and either**

- the condition described in [TPS_SWCT_01851] concerning the referenced `VariableDataPrototype` is not fulfilled **or**

- attribute `RPortPrototype.requiredComSpec.maxNoNewOrRepeatedData` does not exist,

then attribute `EndToEndProtection.endToEndProfile.maxNoNewOrRepeatedData` **shall exist at the time when the contract phase generation is executed**.

⌋*()*

[constr_1216] Existence of attribute `EndToEndDescription.syncCounterInit` for `PROFILE_01` ⌈If the value of attribute `EndToEndDescription.category` is set to `PROFILE_01` **and either**

- the condition described in [TPS_SWCT_01852] concerning the referenced `VariableDataPrototype` is not fulfilled **or**

- attribute `RPortPrototype.requiredComSpec.syncCounterInit` does not exist,

then the attribute `EndToEndProtection.endToEndProfile.syncCounterInit` **shall exist at the time when the contract phase generation is executed**.

⌋*()*

[constr_1217] Existence of attribute `EndToEndDescription.maxNoNewOrRepeatedData` for `PROFILE_02` ⌈If the value of attribute `EndToEndDescription.category` is set to `PROFILE_02` **and either**

- the condition described in [TPS_SWCT_01851] concerning the referenced `VariableDataPrototype` is not fulfilled **or**

- attribute `RPortPrototype.requiredComSpec.maxNoNewOrRepeatedData` does not exist,

then attribute `EndToEndProtection.endToEndProfile.maxNoNewOrRepeated-Data` **shall exist at the time when the contract phase generation is executed**.

⌋*()*

[constr_1218] Existence of attribute `EndToEndDescription.syncCounterInit` for `PROFILE_02` ⌈If the value of attribute `EndToEndDescription.category` is set to `PROFILE_01` **and either**

- the condition described in [TPS_SWCT_01852] concerning the referenced `VariableDataPrototype` is not fulfilled **or**

- attribute `RPortPrototype.requiredComSpec.syncCounterInit` does not exist,

then the attribute `EndToEndProtection.endToEndProfile.syncCounterInit` **shall exist at the time when the contract phase generation is executed**.

⌋*()*

[constr_1219] Invalidation depends on the value of `swImplPolicy` ⌈If the value of `swImplPolicy` of a `SenderReceiverInterface.dataElement` is set to the value `SwImplPolicyEnum.queued`, then the enclosing `SenderReceiverInterface` shall not aggregate in the role `invalidationPolicy` an `InvalidationPolicy`

- that references the `dataElement` and

- where the value of `InvalidationPolicy.handleInvalid` is set to anything else than `HandleInvalidEnum.dontInvalidate`

at the time when the contract phase generation is executed.

⌋*()*

[constr_1220] Compatibility of `SwBaseType` ⌈Two `SwBaseType`s are compatible **at the time when the RTE is generated** if and only if attributes

- `baseTypeSize` respectively

- `byteOrder`,

- `memAlignment`,

- `baseTypeEncoding`, and

- `nativeDeclaration`

have identical values **at the time when the RTE is generated**.

⌋*()*

[constr_1221] `DataPrototype` is typed by an `ApplicationPrimitive-DataType` ⌈If a `DataPrototype` is typed by an `ApplicationPrimitive-DataType`, its `initValue` shall be provided by an `ApplicationValueSpecification`.

If the underlying `ApplicationPrimitiveDataType` represents an enumeration, the value provided shall match to one of the applicable text values (`vt`, `shortLabel`, `symbol`) defined by the applicable `CompuScale`s.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1222] `category` of an `AutosarDataType` used to type a `DataPrototype` is set to `STRING` ⌈If the `category` of an `AutosarDataType` used to type a `DataPrototype` is set to `STRING`, the `ApplicationValueSpecification` used to initialize the `DataPrototype` shall be of `category STRING`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1223] `DataPrototype` is typed by an `ApplicationRecordDataType` ⌈If a `DataPrototype` is typed by an `ApplicationRecordDataType`, the corresponding `initValue` shall be provided by a `RecordValueSpecification`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1224] `DataPrototype` is typed by an `ApplicationArrayDataType` ⌈If a `DataPrototype` is typed by an `ApplicationArrayDataType`, the corresponding `initValue` shall be provided by an `ArrayValueSpecification` (that may contain an `ApplicationRuleBasedValueSpecification`).

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1225] `DataPrototype` is typed by an `ImplementationDataType` that references a `CompuMethod` of category `TEXTTABLE` or `BITFIELD_TEXTTABLE` ⌈If a `DataPrototype` is typed by an `ImplementationDataType` that references a `CompuMethod` of category `TEXTTABLE` or `BITFIELD_TEXTTABLE` the applicable `ValueSpecification` shall be a `TextValueSpecification`.

In this case the value provided shall match to one of the applicable text values (`vt`, `shortLabel`, `symbol`) defined by the applicable `CompuScales`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1226] Applicable range for `ExecutableEntityActivationReason.bitPosition` ⌈The value of attribute `ExecutableEntityActivationReason.bitPosition` shall be in the range of 0 .. 31 **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1227] Value of attribute `ExecutableEntityActivationReason.bitPosition` shall be unique ⌈The value of attributes `ExecutableEntityActivationReason.bitPosition` and `ExecutableEntityActivationReason.symbol` shall be unique in the context of the enclosing `RunnableEntity` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1228] `RTEEvent` that is referenced by a `WaitPoint` in the role `trigger` shall not reference `ExecutableEntityActivationReason` ⌈An `RTEEvent` that is referenced by a `WaitPoint` in the role `trigger` shall not reference `ExecutableEntityActivationReason` in the role `activationReasonRepresentation` **at the time when the RTE is generated**.

⌋*()*

[constr_1229] `category` of `ImplementationDataType` boils down to `VALUE` ⌈An `ImplementationDataType` qualifies as an `Integral Primitive Type` if and only if either

- its `category` is `VALUE` or `TYPE_REFERENCE` that eventually boils down to `VALUE` **or**

- its `category` is `ARRAY` **and** it has only one `subElement` **and** one of the following conditions applies:

  - `subElement.category` is set to `VALUE` or `TYPE_REFERENCE` that eventually boils down to `VALUE` **and** the `subElement` refers to a `SwBaseType` where `baseTypeSize` is set to the value 8 **and** the `baseTypeEncoding` is set to `NONE`.

  - `subElement.category` is set to `TYPE_REFERENCE` **and** the `swDataDefProps.implementationDataType` literally represents the `Platform Data Type` named "uint8".

  - `subElement.category` is set to `TYPE_REFERENCE` **and** the attribute `swDataDefProps.implementationDataType.shortName` is set to "uint8"

and `swDataDefProps.baseType.baseTypeDefinition.nativeDeclaration` does not exist.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1230] `ApplicationDataType` **that qualifies for** `Integral Primitive Type` ⌈An `ApplicationDataType` qualifies as an `Integral Primitive Type` if and only if **all** the following conditions apply:

- `ApplicationDataType.category` is set to `BOOLEAN`, `VALUE`, `STRING`, or `ARRAY`

- in the applicable scope a `DataTypeMap` is available that refers to the given `ApplicationDataType`

- the found `DataTypeMap` refers to an `ImplementationDataType` that fulfills the requirements of [constr_1229]

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1231] `ConsistencyNeeds` **aggregated by** `CompositionSwComponentType` ⌈If `ConsistencyNeeds` are aggregated by a `CompositionSwComponentType` the associations stereotyped ≪instanceRef≫ may only refer to context and target elements within the context of this `CompositionSwComponentType`.

This rule shall be imposed **at the time when the creation of the CompositionSwComponentType is finished**.

⌋*()*

[constr_1232] `ConsistencyNeeds` **aggregated by** `AtomicSwComponentType` ⌈If `ConsistencyNeeds` are aggregated by a `AtomicSwComponentType` the associations stereotyped ≪instanceRef≫ may only refer to context and target elements within the context of this `AtomicSwComponentType` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1233] `InstantiationTimingEventProps` **shall only reference** `TimingEvent` ⌈**at any time in the workflow**, an `InstantiationTimingEventProps` shall only reference `TimingEvent` in the role `refinedEvent`. A reference to other kinds of `RTEEvent`s is not supported.

⌋*()*

[constr_1234] Value of `RunnableEntity.symbol` ⌈The value of a `RunnableEntity.symbol` owned by an `NvBlockSwComponentType` that is triggered by an `Op-`

`erationInvokedEvent` shall only be taken from the set of API names associated with the `NvM`.

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_1237] Scope of mapped `ClientServerOperation`s in the context of a `ClientServerOperationMapping` ⌈All `ClientServerOperation`s referenced by a `ClientServerOperationMapping` in the role `firstOperation` shall belong to exactly one `ClientServerInterface`.

All `ClientServerOperation`s referenced by a `ClientServerOperationMapping` in the role `secondOperation` shall belong to exactly one other `ClientServerInterface`.

This rule shall be imposed **at any time in the workflow**.

⌋*()*

[constr_1238] Scope of mapped `ApplicationError`s in the context of a `ClientServerOperationMapping` ⌈All `ApplicationError`s referenced by a `ClientServerApplicationErrorMapping` in the role `firstApplicationError` shall belong to exactly one `ClientServerInterface`.

All `ApplicationError`s referenced by a `ClientServerApplicationErrorMapping` in the role `secondApplicationError` shall belong to exactly one other `ClientServerInterface`.

This rule shall be imposed **at any time in the workflow**.

⌋*()*

[constr_1240] Consistency of `ArgumentDataPrototype`s within the context of a `ClientServerOperationMapping` ⌈Unless a `ClientServerOperationMapping`.`firstToSecondDataTransformation` exists, for each `argument` owned by

- a `ClientServerOperationMapping`.`firstOperation` and
- `ClientServerOperationMapping`.`secondOperation`,

a reference in the role

- `ClientServerOperationMapping`.`argumentMapping`.`firstDataPrototype` or
- `ClientServerOperationMapping`.`argumentMapping`.`secondDataPrototype`

shall exist **at any time in the workflow**, originated by one of the `ClientServerOperationMapping.argumentMapping`s owned by the mentioned `ClientServerOperationMapping`.

⌋*()*

[constr_1241] `Compound Primitive Data Type`**s and** `invalidValue` ⌈Compound `Primitive Data Type`s that have set the value of `category` other than `STRING` shall **not** define `invalidValue` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1242] **Restriction of** `invalidValue` **for** `ApplicationPrimitiveDataType` **of** `category STRING` ⌈`invalidValue` for `ApplicationPrimitiveDataType` of `category STRING` ([constr_1241] applies) is restricted to be either a compatible `ApplicationValueSpecification` or a `ConstantReference` that in turn points to a compatible `ApplicationValueSpecification` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1243] `NumericalOrText` **shall either define** `vf` **or** `vt` ⌈Within the context of one `NumericalOrText`, **either** the attribute `vf` **or** the attribute `vt` shall be defined. The existence of both attributes at the same time is not permitted.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1244] `DataPrototype`**s used in application software shall not be typed by C enums** ⌈A `ImplementationDataType` that is used to type a `DataPrototype` owned by an `AtomicSwComponentType` shall not set `swDataDefProps`.`additionalNativeTypeQualifier` to `enum` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1245] Consideration of `ModeTransition`**s for the compatibility of** `ModeDeclarationGroup`**s** ⌈One of the following conditions for the consideration of `ModeTransition`s for the compatibility of `ModeDeclarationGroup`s shall apply **at the time when the RTE is generated**:

- **Either** the mode provider **or** the mode user define `ModeTransition`s.

- The `ModeTransition`s defined in the context of the mode provider are **identical** to the `ModeTransition`s defined in the context of the mode user **or** a `ModeDeclarationMapping` mapping is applied.

⌋*()*

[constr_1246] Consistency of `firstMode` and `secondMode` in the scope of one `ModeDeclarationMappingSet` ⌈Within the scope of one `ModeDeclarationMappingSet`,

- all `firstMode`s shall belong to one and only one `ModeDeclarationGroup` and

- all `secondMode`s shall belong to one and only one **other** `ModeDeclarationGroup`

at any time in the workflow.

⌋*()*

[constr_1247] Consistency of `ModeDeclarationMappingSet` with respect to the referenced `firstModeGroup` and `secondModeGroup` ⌈If a `ModeDeclarationGroupPrototypeMapping.modeDeclarationMappingSet` exists, then

- the `ModeDeclarationGroup` owning the `modeDeclaration`s referenced in the role `firstMode` shall be the `type` of the `ModeDeclarationGroupPrototypeMapping.firstModeGroup` and

- the `ModeDeclarationGroup` owning the `modeDeclaration`s referenced in the role `secondMode` shall be the `type` of the `ModeDeclarationGroupPrototypeMapping.secondModeGroup`.

This rule shall be imposed **at any time in the workflow**.

⌋*()*

[constr_1248] Compatibility of `PortPrototype`s of different `DataInterface`s in the context of a `PassThroughSwConnector` ⌈`PortPrototype`s of different `DataInterface`s are considered compatible **at the time when the RTE is generated** if and only if

1. For **at least one** `VariableDataPrototype` or `ParameterDataPrototype` defined in the context of the `DataInterface` of the required outer `PortPrototype` a compatible `VariableDataPrototype` or `ParameterDataPrototype` exists in the `DataInterface` of the provided outer `PortPrototype`.

   Either the `shortName` of `VariableDataPrototype`s and `ParameterDataPrototype`s are used to identify the pair **or** a `PortInterfaceMapping` exists that defines which differently named elements of `PortInterface`s correlate with each other.

2. For each such pair, the values of the `PortInterface.isService` attributes are identical.

⌋*()*

[constr_1249] Compatibility of `ModeSwitchInterface`s in the context of a `PassThroughSwConnector` ⌈`PortPrototype`s of different `ModeSwitchInterface`s are considered compatible **at the time when the RTE is generated** if and only if

1. For the `ModeDeclarationGroupPrototype` defined in the context of the `ModeSwitchInterface` of the required outer `PortPrototype` a compatible `ModeDeclarationGroupPrototype` exists in the `ModeSwitchInterface` of the provided outer `PortPrototype`.

   Either the `shortName`s of the `ModeDeclarationGroupPrototype`s are used to identify the pair **or** a `ModeInterfaceMapping` exists that maps the corresponding `ModeDeclarationGroupPrototype`s.

2. For each such pair, the values of the `PortInterface.isService` attributes are identical.

⌋*()*

[constr_1250] Compatibility of `ClientServerInterface`s in the context of a `PassThroughSwConnector` ⌈`PortPrototype`s of different `ClientServerInterface`s are considered compatible **at the time when the RTE is generated** if and only if

1. For **at least one** `ClientServerOperation` defined in the context of the `ClientServerInterface` of the provided outer `PortPrototype` a compatible `ClientServerOperation` exists in the `ClientServerInterface` of the required outer `PortPrototype`.

   Either the `shortName`s of the `ClientServerOperation`s are used to identify the pair **or** a `ClientServerInterfaceMapping` exists that maps the corresponding `ClientServerOperation`s.

2. For each such pair, the values of the `PortInterface.isService` attributes are identical.

⌋*()*

[constr_1251] Compatibility of `PortPrototype`s of `TriggerInterface`s in the context of a `PassThroughSwConnector` ⌈`PortPrototype`s of different `TriggerInterface`s are considered compatible **at the time when the RTE is generated** if and only if

1. For **at least one** `Trigger` defined in the context of the `TriggerInterface` of the required outer `PortPrototype` a compatible `Trigger` exists in the `TriggerInterface` of the provided outer `PortPrototype`.

   Either the `shortName` of `Trigger`s are used to identify the pair **or** a `TriggerInterfaceMapping` exists that refers to one of the `Trigger`s in the role `firstTrigger` and to the other in the role `secondTrigger`.

2. For each such pair, the values of the `PortInterface.isService` attributes are identical.

⌋*()*

[constr_1252] Creation of a loop involving a `PassThroughSwConnector` is not allowed ⌈**at any time in the workflow**, a `PassThroughSwConnector` is not allowed if the required outer `PortPrototype` is directly or indirectly connected to the provided outer `PortPrototype` without the placement of a `SwComponentPrototype` typed by an `AtomicSwComponentType` in the chain of `SwConnector`s.

⌋*()*

[constr_1253] Allowed multiplicities for attributes of `VariationPointProxy` depending on the applicable binding time and the value of `VariationPointProxy`.`category` ⌈

BindingTime	`category`	Allowed Attribute Multiplicity
PreBuild	VALUE	`valueAccess` [1], `implementationDataType` [0..1]
	CONDITION	`conditionAccess` [1]
PostBuild	VALUE	`postBuildValueAccess` [1], `implementationDataType` [1]
	CONDITION	`postBuildVariantCondition` [1..*], `conditionAccess` [0..1]

⌋*()*

[constr_1254] Definition of a pointer to a pointer ⌈AUTOSAR does **not** support the definition of a pointer to a pointer by defining an `ImplementationDataType` of `category DATA_REFERENCE` that aggregates `SwDataDefProps` in the role `swDataDefProps` that in turn aggregate `SwPointerTargetProps` in the role `swPointerTargetProps` with attribute `targetCategory` set to `DATA_REFERENCE` that in turn aggregates `SwDataDefProps` in the role `swDataDefProps` that aggregates `SwPointerTargetProps` in the role `swPointerTargetProps` that references an `ImplementationDataType` of `category` e.g. `VALUE`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1255] `ApplicationPrimitiveDataType`s of `category BOOLEAN` and `STRING` ⌈If a `Unit` is referenced from within `SwDataDefProps` and/or `PhysConstrs` owned by an `ApplicationPrimitiveDataType`s of `category BOOLEAN` and `STRING` it is required that this `Unit` represents a meaningless unit, i.e. the referenced `physicalDimension` shall not define any exponent value other than 0 **at any time in the workflow**.

⌋*()*

[constr_1256] Acknowledgement feedback in n:1 writer case ⌈Within the scope of one `SwcInternalBehavior`, it is **not** allowed that two or more aggregated `RunnableEntity`s own either `dataSendPoint`s or `dataWriteAccess`s that in turn point to the identical `accessedVariable.autosarVariable.targetDataPrototype` **if** the attribute `transmissionAcknowledge` exists in the context of the `SenderComSpec` owned by the `dataSendPoint.accessedVariable.autosarVariable.portPrototype` (or the respective construct for `dataWriteAccess`) that also refers to said `dataElement`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1257] **No `WaitPoint`s allowed** ⌈A `RunnableEntity` referenced by an `InitEvent` in the role `startOnEvent` shall not aggregate a `WaitPoint` **at the time when the RTE is generated**.

⌋*()*

[constr_1258] **Value of `minimumStartInterval` for `RunnableEntity`s triggered by an `InitEvent`** ⌈The value of the attribute `ExecutableEntity.minimumStartInterval` for a `RunnableEntity`s that is triggered by an `InitEvent` shall always be set to 0 **at the time when the RTE is generated**.

⌋*()*

[constr_1259] **Aggregation of `AsynchronousServerCallPoint` and `AsynchronousServerCallResultPoint`** ⌈A `RunnableEntity` referenced by an `InitEvent` in the role `startOnEvent` may aggregate an `AsynchronousServerCallPoint` but it shall not aggregate an `AsynchronousServerCallResultPoint` **at the time when the RTE is generated**.

⌋*()*

[constr_1260] **No mode disabling for `InitEvent`s** ⌈An `InitEvent` shall not have a reference to a `ModeDeclaration` in the role `disabledMode` **at the time when the RTE is generated**.

⌋*()*

[constr_1261] **Applicability for `EndToEndDescription.dataIdNibbleOffset`** ⌈`EndToEndDescription.dataIdNibbleOffset` shall be used **only** if `EndToEndDescription.dataIdMode` is set to the value 3 **and** at the same time `EndToEndDescription.category` is set to `PROFILE_01`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1263] **Existence of `ModeErrorBehavior.defaultMode`** ⌈The optional attribute `ModeErrorBehavior.defaultMode` **shall exist** if the value of the attribute `ModeErrorBehavior.errorReactionPolicy` is set to `defaultMode`.

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_1268] **`ArgumentDataPrototype.direction` shall be preserved in a `ClientServerOperationMapping`** ⌈Within the context of a `ClientServerOperationMapping`, the value of the argument `ArgumentDataPrototype.direction`

of two mapped `ArgumentDataPrototype` shall be identical **at any time in the workflow**.

⌋*()*

[constr_1269] Number of `arguments` shall be preserved in a `ClientServerOperationMapping` ⌈Within the context of a `ClientServerOperationMapping`, the number of `argument`s of `firstOperation` and `secondOperation` shall be identical **at any time in the workflow**.

⌋*()*

[constr_1270] `ArgumentDataPrototype` shall be mapped only once in a `ClientServerOperationMapping` ⌈Within the context of a `ClientServerOperationMapping`, each `argument` shall only be referenced **once** in the role `firstDataPrototype` or `secondDataPrototype` **at any time in the workflow**.

⌋*()*

[constr_1271] `RecordValueSpecification.field`s shall be identical to the number of `ApplicationRecordDataType.element`s ⌈The initialization of a `DataPrototype` typed by an `ApplicationRecordDataType` by means of a `RecordValueSpecification` shall exactly match the structure of the `ApplicationRecordDataType`.

For this means, it is required that the number of `RecordValueSpecification.field`s shall be identical to the number of `ApplicationRecordDataType.element`s.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1272] `RecordValueSpecification.field`s shall be identical to the number of `subElement`s of `ImplementationDataType` of `category STRUCTURE` ⌈The initialization of an `DataPrototype` typed by an `ImplementationDataType` of `category STRUCTURE` by means of a `RecordValueSpecification` shall exactly match the structure of the `ImplementationDataType` of `category STRUCTURE`.

For this means, it is required that the number of `RecordValueSpecification.field`s shall be identical to the number of `ImplementationDataType.subElement`s.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1273] Rules for the initialization of `ApplicationArrayDataType` by means of `ArrayValueSpecification` ⌈The following rules apply for the initializa-

tion of a `DataPrototype` typed by an `ApplicationArrayDataType` by means of an `ArrayValueSpecification`:

- If the attribute `ApplicationArrayDataType.element.arraySizeSemantics` is set to **`fixedSize`** then the `ArrayValueSpecification` shall exactly match the structure of the `ApplicationArrayDataType`.

  This means that the number of `ArrayValueSpecification.element`s shall be identical to the value of `ApplicationArrayDataType.element.maxNumberOfElements`.

- If the attribute `ApplicationArrayDataType.element.arraySizeSemantics` is set to **`variableSize`** and the `ArrayValueSpecification` **does not define** attribute `intendedPartialInitializationCount` then `ArrayValueSpecification` shall **exactly** match the structure of the `ApplicationArrayDataType`.

  This means that the number of `ArrayValueSpecification.element`s shall be identical to the value of `ApplicationArrayDataType.element.maxNumberOfElements`.

- If the attribute `ApplicationArrayDataType.element.arraySizeSemantics` is set to **`variableSize`** and the `ArrayValueSpecification` specifies a value for attribute `intendedPartialInitializationCount` then `ArrayValueSpecification` shall contain **exactly** `intendedPartialInitializationCount` elements.

  This includes the case that the value of `intendedPartialInitializationCount` is set to 0 (i.e. "empty" initialization) and the case that the `intendedPartialInitializationCount` is set to the value of the respective `ApplicationArrayElement.maxNumberOfElements` (i.e. "full" initialization).

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1274] Rules for the initialization of array-shaped `ImplementationDataType` with a fixed size by means of `ArrayValueSpecification` ⌈The following rule applies for the initialization of a `DataPrototype` typed by an `ImplementationDataType` of `category` ARRAY where attribute `ImplementationDataType.subElement.arraySizeSemantics` is set to `fixedSize` by means of an `ArrayValueSpecification`: the `ArrayValueSpecification` shall exactly match the structure of the `ImplementationDataType`.

This means that the number of `ArrayValueSpecification.element`s shall be identical to the value of `ImplementationDataType.subElement.arraySize`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1277] SwDataDefProps.swImplPolicy of a VariableDataPrototype referenced by a VariableAccess aggregated in the role dataReceivePoint-ByValue ⌈The SwDataDefProps.swImplPolicy of a VariableDataPrototype referenced by a VariableAccess aggregated in the role dataReceivePoint-ByValue shall not be set to queued **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1278] PhysConstrs references a Unit ⌈DataConstrs are only compatible if the DataConstr.dataConstrRule.physConstrs.unit are compatible or neither DataConstr.dataConstrRule.physConstrs.unit exist **at the time when the RTE is generated**.

⌋*()*

[constr_1279] Unmapped elements of ApplicationCompositeDataTypes or ImplementationDataTypes and the attribute swImplPolicy ⌈If the attribute swImplPolicy is set to queued, then it is not allowed to have unmapped elements of ApplicationCompositeDataTypes or ImplementationDataTypes of category STRUCTURE or ARRAY on the "target" end **at the time when the RTE is generated**.

⌋*()*

[constr_1280] Unmapped dataElement on the "target" end shall have an initValue ⌈If elements of ApplicationCompositeDataTypes or ImplementationDataTypes of category STRUCTURE or ARRAY are not considered in a SubElementMapping **and** the NonqueuedReceiverComSpec is aggregated by an AbstractRequiredPortPrototype referenced by the "target" end, then the enclosing dataElement shall have an initValue.

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_1282] Restriction concerning the usage of RuleBasedValueSpecification or a ReferenceValueSpecification for the specification of an invalidValue ⌈The aggregation of a RuleBasedValueSpecification or a ReferenceValueSpecification for the definition of a ApplicationPrimitiveDataType.swDataDefProps.invalidValue is not supported **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1284] Limitation of the use of TextValueSpecification ⌈TextValueSpecification shall **only** be used in the context of an AutosarDataType that references a CompuMethod in the role ImplementationDataType.swDataDefProps.compuMethod of category TEXTTABLE and BITFIELD_TEXTTABLE.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1285] Applicability of roles vs. `PortPrototype`s ⌈The aggregation of `AutosarVariableRef` aggregated by `NvBlockDataMapping` in the roles `writtenNvData`, `writtenReadNvData`, or `readNvData` is subject to limitation **at the time when the RTE is generated**, depending on the applicable subclass of `PortPrototype`:

- The role `writtenNvData` shall only be used if the corresponding `PortPrototype` is a `RPortPrototype`

- The role `writtenReadNvData` shall only be used if the corresponding `PortPrototype` is a `PRPortPrototype`

- The role `readNvData` shall only be used if the corresponding `PortPrototype` is a `PPortPrototype`

⌋*()*

[constr_1286] `serverArgumentImplPolicy` and `ArgumentDataPrototype` typed by primitive data types ⌈The value of the attribute `ArgumentDataPrototype.serverArgumentImplPolicy` shall **not** be set to `useVoid` for an `ArgumentDataPrototype` of `direction in` that is typed by an `AutosarDataType` that boils down to a primitive C data type (see [TPS_SWCT_01565]).

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1287] Compatibility of `SenderReceiverInterface`s with respect to `invalidationPolicy` ⌈`VariableDataPrototype`s defined in the context of the `SenderReceiverInterface` are only compatible if the `invalidationPolicy`s have the same value.

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_1288] Allowed Attributes vs. `category` for `DataPrototype`s typed by `ImplementationDataType`s ⌈

Attributes of SwDataDefProps	Root Element			Attribute Existence per Category						
	DataPrototype	InstantiationDataDefProps	ParameterAccess	VALUE	DATA_REFERENCE	FUNCTION_REFERENCE	TYPE_REFERENCE	STRUCTURE	UNION	ARRAY
additionalNativeTypeQualifier										
annotation	x	x	*	*	*	*	*	*	*	*
baseType										
compuMethod										
dataConstr.dataConstrRule.physConstrs	x	x		d/c[13]			d/c			d/c
dataConstr.dataConstrRule.internalConstrs	x	x		0..1			0..1			0..1
displayFormat	x	x		0..1			0..1	0..1	0..1	0..1
displayPresentation	x	x		0..1			0..1			0..1
implementationDataType										
invalidValue										
stepSize	x	x		0..1						0..1
swAddrMethod	x	x		0..1	0..1	0..1	0..1	0..1	0..1	0..1
swAlignment	x	x		0..1	0..1	0..1	0..1	0..1	0..1	0..1
swBitRepresentation										
swCalibrationAccess	x	x		0..1			0..1	0..1	0..1	0..1
swCalprmAxisSet										
swComparisonVariable										
swDataDependency										
swHostVariable										
swImplPolicy	x			0..1	0..1	0..1	0..1	0..1	0..1	0..1
swIntendedResolution										
swInterpolationMethod										
swIsVirtual										
swPointerTargetProps										
swPointerTargetProps.swDataDefProps										
swPointerTargetProps.functionPointerSignature										
swRecordLayout										
swRefreshTiming	x	x		0..1			0..1	0..1	0..1	0..1
swTextProps										
swValueBlockSize										
swValueBlockSizeMult										
unit										
valueAxisDataType										

⌋*()*

---

[13]don't care

[constr_1289] Allowed Attributes vs. `category` for `DataPrototype`s typed by `ApplicationDataType`s

Attributes of SwDataDefProps	Root El.			Attribute Existence per Category												
	DataPrototype	InstantiationDataDefProps	ParameterAccess	VALUE	VAL_BLK	STRUCTURE	ARRAY	STRING	BOOLEAN	COM_AXIS	RES_AXIS	CURVE	MAP	CUBOID	CUBE_4	CUBE_5
additionalNativeTypeQualifier																
annotation	x	x	x	*	*	*	*	*	*	*	*	*	*	*	*	*
baseType																
compuMethod																
dataConstr.dataConstrRule.physConstrs	x	x		0..1	0..1		0..1		0..1			0..1	0..1	0..1	0..1	0..1
dataConstr.dataConstrRule.internalConstrs	x	x		d/c[14]	d/c		d/c		d/c			d/c	d/c	d/c	d/c	d/c
displayFormat	x	x		0..1	0..1		0..1	0..1	0..1			0..1	0..1	0..1	0..1	0..1
displayPresentation	x	x		0..1	0..1		0..1			0..1	0..1	0..1	0..1	0..1	0..1	0..1
implementationDataType																
invalidValue																
stepSize	x	x	x	0..1	0..1		0..1			0..1	0..1	0..1	0..1	0..1	0..1	0..1
swAddrMethod	x	x		0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1
swAlignment	x	x		0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1
swBitRepresentation																
swCalibrationAccess	x	x		0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1
swCalprmAxisSet																
swCalprmAxisSet.swCalprmAxis/SwAxisGrouped.swCalprmRef		x	x				0..1					0..1	0..1	0..1	0..1	0..1
swCalprmAxisSet.swCalprmAxis/SwAxisIndividual.swVariableRef		x	x				0..1			0..1	0..1	0..1	0..1	0..1	0..1	0..1
swCalprmAxisSet.swCalprmAxis/SwAxisGrouped.sharedAxisType																
swCalprmAxisSet.swCalprmAxis/SwAxisIndividual.inputVariableType																
swCalprmAxisSet.swCalprmAxis/SwAxisIndividual.unit																
swComparisonVariable			x									0..1	0..1	0..1	0..1	0..1
swDataDependency	x	x		0..1								0..1	0..1	0..1	0..1	0..1
swHostVariable																
swImplPolicy	x			0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1	0..1
swIntendedResolution																
swInterpolationMethod	x	x	x	0..1						0..1	0..1	0..1	0..1	0..1	0..1	0..1
swIsVirtual	x	x		0..1					0..1			0..1	0..1	0..1	0..1	0..1
swPointerTargetProps																

▽

---

[14]don't care

$\triangle$

Attributes of SwDataDefProps	Root El.			Attribute Existence per Category												
	DataPrototype	InstantiationDataDefProps	ParameterAccess	VALUE	VAL_BLK	STRUCTURE	ARRAY	STRING	BOOLEAN	COM_AXIS	RES_AXIS	CURVE	MAP	CUBOID	CUBE_4	CUBE_5
swRecordLayout																
swRefreshTiming	x	x		0..1	0..1			0..1	0..1							
swTextProps																
swValueBlockSize																
swValueBlockSizeMult																
unit																
valueAxisDataType																

⌋*()*

[constr_1290] Limitation on the number of `PPortComSpec`s in the context of one `PPortPrototype` ⌈Within the context of one `PPortPrototype`, there can only be **one** (sub-class of) `PPortComSpec` that references a given

- `dataElement` (in the case of `SenderComSpec`),
- `operation` (in the case of `ServerComSpec`),
- `modeGroup` (in the case of `ModeSwitchSenderComSpec`),
- `variable` (in the case of `NvProvideComSpec`), or
- `parameter` (in the case of `ParameterProvideComSpec`)

at the time when the contract phase generation is executed.

⌋*()*

[constr_1291] Limitation on the number of `RPortComSpec`s in the context of one `PPortPrototype` ⌈Within the context of one `RPortPrototype`, there can only be **one** `RPortComSpec` that references a given

- `dataElement` (in the case of `ReceiverComSpec`),
- `operation` (in the case of `ClientComSpec`),
- `modeGroup` (in the case of `ModeSwitchReceiverComSpec`),
- `variable` (in the case of `NvRequireComSpec`), or
- `parameter` (in the case of `ParameterRequireComSpec`)

at the time when the contract phase generation is executed.

⌋*()*

[constr_1292] Limitation on the number of RPortComSpecs/PPortComSpecs in the context of one PRPortPrototype ⌈Within the context of one `PRPortPrototype`, there can only be **one** `RPortComSpec` and **one** `PPortComSpec` that references a given

- `dataElement` (in the case of `ReceiverComSpec`/`SenderComSpec`),

- `operation` (in the case of `ClientComSpec`/`ServerComSpec`),

- `modeGroup` (in the case of `ModeSwitchReceiverComSpec`/`ModeSwitchSenderComSpec`), or

- `variable` (in the case of `NvRequireComSpec`/`NvProvideComSpec`)

at the time when the contract phase generation is executed.

⌋*()*

[constr_1295] PortInterfaces and category DATA_REFERENCE ⌈A `DataPrototype` defined in the context of a `PortInterface` used by an `ApplicationSwComponentType` or `SensorActuatorSwComponentType` that is (after potential indirections via `TYPE_REFERENCE` are resolved) either typed by or mapped to an `ImplementationDataType` of category `DATA_REFERENCE` shall only be used if either the provider or the requester of the information represents a `ServiceSwComponentType`, a `ComplexDeviceDriverSwComponentType`, a `ParameterSwComponentType`, or an `NvBlockSwComponentType`, or the `EcuAbstractionSwComponentType`.

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_1296] DataPrototypes used as explicitInterRunnableVariable or implicitInterRunnableVariable and category DATA_REFERENCE ⌈A `VariableDataPrototype` shall not be aggregated by `SwcInternalBehavior` in either the role:

- `explicitInterRunnableVariable`, or

- `implicitInterRunnableVariable`

if the `VariableDataPrototype` (after potential indirections via `TYPE_REFERENCE` are resolved) is either typed by, or mapped to, an:

- `ImplementationDataType` of category `DATA_REFERENCE`, or

- `ImplementationDataType` that contains `subElement`s that (after potential indirections via `TYPE_REFERENCE` are resolved) are of category `DATA_REFERENCE`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1298] Existence of attributes if `category` of a `ModeDeclarationGroup` is set to `EXPLICIT_ORDER` ⌈The attributes `ModeDeclarationGroup.onTransitionValue` and `ModeDeclaration.value` (for each `ModeDeclaration`) shall be set if the `category` of a `ModeDeclarationGroup` is set to `EXPLICIT_ORDER`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1299] Existence of attributes if `category` of a `ModeDeclarationGroup` is set to other than `EXPLICIT_ORDER` ⌈The attributes `ModeDeclarationGroup.onTransitionValue` or `ModeDeclaration.value` (for any `ModeDeclaration`) shall **not** be set if the `category` of a `ModeDeclarationGroup` is set to any value **other than** `EXPLICIT_ORDER` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1300] Primitive `DataPrototype` on the "source" end shall not be mapped to element of a composite data type on the "target" end of the `SwConnector` ⌈The usage of `DataPrototypeMapping` or `SubElementMapping` does not support the following configuration:

- The `AutosarDataPrototype` contained in the `PortPrototype` on the "source" end of an `SwConnector` is typed by an `ApplicationPrimitiveDataType` of category VALUE or `ImplementationDataType` of category VALUE or category `TYPE_REFERENCE` that eventually resolves to category VALUE.

- The `DataPrototypeMapping` aggregates a `subElementMapping` that refers to a `ImplementationDataTypeElement` or `ApplicationCompositeElementDataPrototype` contained in the `PortPrototype` on the "target" end.

This rule shall be imposed **at any time in the workflow**.

⌋*()*

[constr_1301] Existence of `RoleBasedDataTypeAssignment.role` vs. `RoleBasedDataAssignment.role` ⌈The usage of a `RoleBasedDataTypeAssignment` with attribute `role` set to the value `temporaryRamBlock` is only allowed if **no** `RoleBasedDataAssignment` defined with attribute `role` set to value `defaultValue` exists in the owning `SwcServiceDependency`.

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_1302] Restriction of data invalidation ⌈Data invalidation is only applicable for one of the following cases applicable on the **receiving** side:

1. `VariableDataPrototype`s typed by either an `ApplicationPrimitive-DataType` or an `ImplementationDataType` of `category VALUE` or `TYPE_-REFERENCE` that boils down to `category VALUE` that have defined an `invalidValue`.

2. `VariableDataPrototype`s typed by either an `ApplicationComposite-DataType` or an `ImplementationDataType` of `category STRUCTURE`, or `ARRAY` or of `category TYPE_REFERENCE` that boils down to `category STRUCTURE`, or `ARRAY` that have **at least one** primitive element with an `invalidValue`.

This rule shall be imposed `at the time when the contract phase generation is executed`.

⌋*()*

[constr_1303] Applicability of `TextTableMapping` depending on the value of `CompuMethod.category` ⌈If − `at the time when the RTE is generated` − a `DataPrototypeMapping` aggregates a `TextTableMapping` then only certain combinations of the value of the applicable `CompuMethod.category` are supported:

- `category` of `firstDataPrototype`: TEXTTABLE,
  `category` of `secondDataPrototype`: TEXTTABLE

- `category` of `firstDataPrototype`: SCALE_LINEAR_AND_TEXTTABLE,
  `category` of `secondDataPrototype`: TEXTTABLE

- `category` of `firstDataPrototype`: TEXTTABLE,
  `category` of `secondDataPrototype`: SCALE_LINEAR_AND_TEXTTABLE

- `category` of `firstDataPrototype`: BITFIELD_TEXTTABLE,
  `category` of `secondDataPrototype`: TEXTTABLE

- `category` of `firstDataPrototype`: TEXTTABLE,
  `category` of `secondDataPrototype`: BITFIELD_TEXTTABLE

- `category` of `firstDataPrototype`: BITFIELD_TEXTTABLE,
  `category` of `secondDataPrototype`: BITFIELD_TEXTTABLE

⌋*()*

[constr_1304] Existence of attribute `bitfieldTextTableMaskFirst` ⌈The attribute `bitfieldTextTableMaskFirst` shall be defined **only if** the `firstDataPrototype` of a `DataPrototypeMapping` refers to a `CompuMethod` that has the value of `category` set to BITFIELD_TEXTTABLE.

This rule shall be imposed `at the time when the RTE is generated`.

⌋*()*

[constr_1305] Existence of attribute `bitfieldTextTableMaskSecond` ⌈The attribute `bitfieldTextTableMaskSecond` shall be defined **only if** the `secondDataPrototype` of a `DataPrototypeMapping` refers to a `CompuMethod` that has the value of `category` set to BITFIELD_TEXTTABLE.

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_1306] Limitation of `TextTableMapping` for `CompuMethod`s that have the value of `category` set to BITFIELD_TEXTTABLE ⌈For any `TextTableMapping` where both `firstDataPrototype` and `secondDataPrototype` refer to `CompuMethod`s that have the value of `category` set to BITFIELD_TEXTTABLE **and** where the attribute `TextTableMapping.valuePair` exists the value of attribute `TextTableMapping.identicalMapping` shall be set to false.

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_1307] Consistency of values and masks in `TextTableMapping` ⌈If a `TextTableMapping` element defines bit masks as `bitfieldTextTableMaskFirst` or `bitfieldTextTableMaskSecond` then all contained `TextTableMapping.valuePair.firstValue`s as well as all `TextTableMapping.valuePair.secondValue`s shall **not** specify a value that would be ruled out when - depending on the given value of `TextTableMapping.mappingDirection` - the relevant bit mask is applied.

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_1308] Existence of `NvBlockNeeds.cyclicWritingPeriod` ⌈The attribute `NvBlockNeeds.cyclicWritingPeriod` shall exist if and only if the attribute `NvBlockNeeds.storeCyclic` exists and its value is set to `true`.

This rule shall be imposed **at the time when the RTE is generated**

⌋*()*

[constr_1309] Existence of `NvBlockDescriptor.timingEvent` ⌈The attribute `NvBlockDescriptor.timingEvent` shall exist **at the time when the RTE is generated** if and only if the `NvBlockDescriptor.nvBlockNeeds.storeCyclic` exists and is set to the value `true`.

⌋*()*

[constr_1310] Existence of attributes of meta-class `NvBlockNeeds` ⌈If in the context of an `ApplicationSwComponentType` the attribute `SwcServiceDependency.serviceNeeds` is implemented by an `NvBlockNeeds` then the following attributes

- `NvBlockNeeds.storeCyclic`

- NvBlockNeeds.cyclicWritingPeriod

- NvBlockNeeds.storeEmergency

- NvBlockNeeds.storeImmediate

- NvBlockNeeds.storeOnChange

shall only exist if in the context of the same SwcServiceDependency a SwcServiceDependency.assignedPort exists that has the attribute role set to the value NvDataPort.

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_1311] Appearance of safety-related possible values of MemorySection.option or SwAddrMethod.option ⌈Any given collection of values stored in the attributes MemorySection.option or SwAddrMethod.option according to [TPS_-SWCT_01456] shall at most include a single value out of the following list **at the time when the RTE is generated**:

- **safetyQM**

- **safetyAsilA**

- **safetyAsilB**

- **safetyAsilC**

- **safetyAsilD**

⌋*()*

[constr_1312] PortPrototypes typed by a ParameterInterface ⌈PortPrototypes typed by a ParameterInterface can either be PPortPrototypes or RPortPrototypes. The usage of PRPortPrototypes that are typed by a ParameterInterface is not supported **at any time in the workflow**.

⌋*()*

[constr_1313] Completeness of TextTableMapping for the values of a given bit mask on the sender side ⌈If a DataPrototypeMapping contains one or more TextTableMapping(s) where the DataPrototype on the **sender side** refers to a CompuMethod of category BITFIELD_TEXTTABLE then all DataPrototypeMapping.textTableMapping shall aggregate a collection of TextTableMapping.valuePair where each possible value of the **sender bit mask**[15] is represented by exactly one TextTableValuePair.firstValue ([TPS_SWCT_01163]) or TextTableValuePair.secondValue ([TPS_SWCT_01164]).

---

[15]Depending on the applicable case this means either bitfieldTextTableMaskFirst (applies if [TPS_SWCT_01163] is in place) or bitfieldTextTableMaskSecond for the case of [TPS_SWCT_-01164].

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_1314] Profile `VSA_LINEAR` for `ApplicationArrayDataType` ⌈If the `dynamicArraySizeProfile` of `ApplicationArrayDataType` is set to `VSA_LINEAR`, the contained `ApplicationArrayElement` shall fulfill **all** the following conditions **at the time when the contract phase generation is executed**:

- The attribute `ApplicationArrayElement.arraySizeSemantics` shall set to the value `variableSize`.

- The attribute `ApplicationArrayElement.maxNumberOfElements` shall be defined.

- The attribute `ApplicationArrayElement.arraySizeHandling` shall be set to the value `allIndicesSameArraySize`.

- The `ApplicationArrayElement` shall be typed by an `ApplicationDataType` that is not an `ApplicationArrayDataType` where the attribute `dynamicArraySizeProfile` exists.

⌋*()*

[constr_1315] Profile `VSA_SQUARE` for `ApplicationArrayDataType` ⌈If the `dynamicArraySizeProfile` of `ApplicationArrayDataType` is set to `VSA_SQUARE`, the contained `ApplicationArrayElement` shall fulfill **all** the following conditions :

- The attribute `ApplicationArrayElement.arraySizeSemantics` shall be set to the value `variableSize`.

- The attribute `ApplicationArrayElement.maxNumberOfElements` shall not be defined.

- The attribute `ApplicationArrayElement.arraySizeHandling` shall be set to the value `inheritedFromArrayElementTypeSize`.

- The `ApplicationArrayElement` shall be typed by an `ApplicationArrayDataType`.

The referred `ApplicationArrayDataType` shall refer over a chain (under consideration of the number of dimensions of the "root" `ApplicationArrayDataType`) of nested `ApplicationArrayDataType`s with `ApplicationArrayElement`s to an `ApplicationDataType` that is **not** an `ApplicationArrayDataType` where the attribute `dynamicArraySizeProfile` exists.

The last `ApplicationArrayDataType` in that chain shall have an `ApplicationArrayElement` that fulfills **all** the following conditions:

- The attribute `ApplicationArrayElement.arraySizeSemantics` shall be set to the value `variableSize`.

- The attribute `ApplicationArrayElement.maxNumberOfElements` shall be defined.

- The attribute `ApplicationArrayElement.arraySizeHandling` set to the value `allIndicesSameArraySize`.

All `ApplicationArrayDataType`s before shall have an `ApplicationArrayElement` that fulfills **all** the following conditions:

- The attribute `ApplicationArrayElement.arraySizeSemantics` shall be set to the value `variableSize`.

- The attribute `ApplicationArrayElement.maxNumberOfElements` shall not be defined.

- The attribute `ApplicationArrayElement.arraySizeHandling` shall be set to the value `inheritedFromArrayElementTypeSize`.

- The `ApplicationArrayElement` shall be typed by an `ApplicationArrayDataType`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1316] Profile `VSA_RECTANGULAR` for `ApplicationArrayDataType` ⌈If the `dynamicArraySizeProfile` of `ApplicationArrayDataType` is set to `VSA_-RECTANGULAR` the contained `ApplicationArrayElement` shall fulfill **all** the following conditions :

- The attribute `ApplicationArrayElement.arraySizeSemantics` shall be set to the value `variableSize`.

- The attribute `ApplicationArrayElement.maxNumberOfElements` shall be defined.

- The attribute `ApplicationArrayElement.arraySizeHandling` shall be set to the value `allIndicesSameArraySize`.

- The `ApplicationArrayElement` shall be typed by an `ApplicationArrayDataType`.

The referred `ApplicationArrayDataType` shall refer over a chain (under consideration of the number of dimensions of the "root" `ApplicationArrayDataType`) of nested `ApplicationArrayDataType`s with `ApplicationArrayElement`s to an `ApplicationDataType` that is **not** an `ApplicationArrayDataType` where the attribute `dynamicArraySizeProfile` exists.

The last `ApplicationArrayDataType` in that chain shall have an `ApplicationArrayElement` that fulfills **all** the following conditions:

- The attribute `ApplicationArrayElement.arraySizeSemantics` shall be set to the value `variableSize`.

- The attribute `ApplicationArrayElement.maxNumberOfElements` shall be defined.

- The attribute `ApplicationArrayElement.arraySizeHandling` shall be set to the value `allIndicesSameArraySize`.

All `ApplicationArrayDataType`s before shall have an `ApplicationArrayElement` that fulfills **all** the following conditions:

- The attribute `ApplicationArrayElement.arraySizeSemantics` shall set to the value `variableSize`

- The attribute `ApplicationArrayElement.maxNumberOfElements` shall be defined.

- The attribute `ApplicationArrayElement.arraySizeHandling` shall be set to the value `allIndicesSameArraySize`.

- The `ApplicationArrayElement` shall be typed by an `ApplicationArrayDataType`.

This rule shall be imposed **vat the time when the contract phase generation is executed**.

⌋*()*

[constr_1317] Profile `VSA_FULLY_FLEXIBLE` for `ApplicationArrayDataType` ⌈If the `dynamicArraySizeProfile` of `ApplicationArrayDataType` is set to `VSA_FULLY_FLEXIBLE`, the contained `ApplicationArrayElement` shall fulfill **all** the following conditions:

- The attribute `ApplicationArrayElement.arraySizeSemantics` shall be set to the value `variableSize`.

- The attribute `ApplicationArrayElement.maxNumberOfElements` shall be defined.

- The attribute `ApplicationArrayElement.arraySizeHandling` shall be set to the value `allIndicesDifferentArraySize`.

- The `ApplicationArrayElement` shall be typed by an `ApplicationArrayDataType`.

The referred `ApplicationArrayDataType` shall refer over a chain (under consideration of the number of dimensions of the "root" `ApplicationArrayDataType`) of nested `ApplicationArrayDataType`s with `ApplicationArrayElement`s to an `ApplicationDataType` that is **not** an `ApplicationArrayDataType` where the attribute `dynamicArraySizeProfile` exist.

The last `ApplicationArrayDataType` in that chain shall have an `Application-ArrayElement` that fulfills **all** the following conditions:

- The attribute `ApplicationArrayElement.arraySizeSemantics` shall be set to the value `variableSize`.

- The attribute `ApplicationArrayElement.maxNumberOfElements` shall be defined.

- The attribute `ApplicationArrayElement.arraySizeHandling` shall be set to the value `allIndicesSameArraySize`.

All `ApplicationArrayDataType`s before shall have an `ApplicationArrayElement` that fulfills **all** the following conditions:

- The attribute `ApplicationArrayElement.arraySizeSemantics` shall be set to the value `variableSize`.

- The attribute `ApplicationArrayElement.maxNumberOfElements` shall be defined.

- The attribute `ApplicationArrayElement.arraySizeHandling` shall be set to the value `allIndicesDifferentArraySize`.

- The `ApplicationArrayElement` shall be typed by an `ApplicationArrayDataType`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1318] Profile `VSA_LINEAR` for `ImplementationDataType` ⌈If the value of attribute `ImplementationDataType.dynamicArraySizeProfile` is set to `VSA_-LINEAR`, the `ImplementationDataType` shall aggregate a `VSA Payload` `ImplementationDataTypeElement` that fulfills all the following conditions:

- The attribute `ImplementationDataTypeElement.arraySizeSemantics` shall not be defined.

- The attribute `ImplementationDataTypeElement.category` shall be set to `ARRAY`.

- The attribute `ImplementationDataTypeElement.arraySize` shall not be defined.

- The attribute `ImplementationDataTypeElement.arraySizeHandling` shall not be defined.

The `VSA Payload` `ImplementationDataTypeElement` shall immediately aggregate another `ImplementationDataTypeElement` that shall fulfill all the following conditions:

- The attribute `ImplementationDataTypeElement.arraySizeSemantics` shall be set to the value `variableSize`.

- The attribute `ImplementationDataTypeElement.arraySize` shall be defined.

- The attribute `ImplementationDataTypeElement.arraySizeHandling` shall be set to the value `allIndicesSameArraySize`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1319] Profile `VSA_SQUARE` for `ImplementationDataType` ⌈If the value of attribute `ImplementationDataType.dynamicArraySizeProfile` is set to `VSA_SQUARE`, the `ImplementationDataType` shall aggregate a `VSA Payload ImplementationDataTypeElement` that fulfills all the following conditions:

- The attribute `ImplementationDataTypeElement.arraySizeSemantics` shall not be defined.

- The attribute `ImplementationDataTypeElement.category` shall be set to the value `ARRAY`.

- The attribute `ImplementationDataTypeElement.arraySize` shall not be defined.

- The attribute `ImplementationDataTypeElement.arraySizeHandling` shall not be defined.

The `VSA Payload ImplementationDataTypeElement` shall immediately aggregate another `ImplementationDataTypeElement` (representing the first dimension) that shall fulfill all the following conditions:

- The attribute `ImplementationDataTypeElement.arraySizeSemantics` shall be set to the value `variableSize`.

- The attribute `ImplementationDataTypeElement.category` shall be set to the value `ARRAY`.

- The attribute `ImplementationDataTypeElement.arraySize` shall not be defined.

- The attribute `ImplementationDataTypeElement.arraySizeHandling` shall be set to the value `inheritedFromArrayElementTypeSize`.

All **intermediate** `ImplementationDataTypeElement`s in the aggregation chain that do not terminate the chain shall fulfill all the following conditions:

- The attribute `ImplementationDataTypeElement.arraySizeSemantics` shall be set to the value `variableSize`.

- The attribute `ImplementationDataTypeElement.category` shall be set to the value `ARRAY`.

- The attribute `ImplementationDataTypeElement.arraySize` shall not be defined.

- The attribute `ImplementationDataTypeElement.arraySizeHandling` shall be set to the value `inheritedFromArrayElementTypeSize`.

The **terminating** `ImplementationDataTypeElement` in the aggregation chain shall fulfill all the following conditions:

- The attribute `ImplementationDataTypeElement.arraySizeSemantics` shall be set to the value `variableSize`.

- The attribute `ImplementationDataTypeElement.arraySize` shall be defined.

- The attribute `ImplementationDataTypeElement.arraySizeHandling` shall be set to the value `allIndicesSameArraySize`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1320] Profile `VSA_RECTANGULAR` for `ImplementationDataType` ⌈If the value of attribute `ImplementationDataType.dynamicArraySizeProfile` is set to `VSA_RECTANGULAR`, the `ImplementationDataType` shall aggregate a `VSA Payload ImplementationDataTypeElement` that fulfills all the following conditions:

- The attribute `ImplementationDataTypeElement.arraySizeSemantics` shall not be defined.

- The attribute `ImplementationDataTypeElement.category` shall be set to the value `ARRAY`.

- The attribute `ImplementationDataTypeElement.arraySize` shall not be defined.

- The attribute `ImplementationDataTypeElement.arraySizeHandling` shall not be defined.

The `VSA Payload ImplementationDataTypeElement` shall immediately aggregate another `ImplementationDataTypeElement` (representing the first dimension) that shall fulfill all the following conditions:

- The attribute `ImplementationDataTypeElement.category` shall be set to the value `ARRAY`.

- The attribute `ImplementationDataTypeElement.arraySizeSemantics` shall be set to the value `variableSize`.

- The attribute ImplementationDataTypeElement.arraySize shall be defined.

- The attribute ImplementationDataTypeElement.arraySizeHandling shall be set to the value allIndicesSameArraySize.

All **intermediate** ImplementationDataTypeElements in the aggregation chain that do not terminate the chain shall fulfill all the following conditions:

- The attribute ImplementationDataTypeElement.category shall be set to the value ARRAY.

- The attribute ImplementationDataTypeElement.arraySizeSemantics shall be set to the value variableSize.

- The attribute ImplementationDataTypeElement.arraySize shall be defined.

- The attribute ImplementationDataTypeElement.arraySizeHandling shall be set to the value allIndicesSameArraySize.

The **terminating** ImplementationDataTypeElement in the aggregation chain shall fulfill all the following conditions:

- The attribute ImplementationDataTypeElement.arraySizeSemantics shall be set to the value variableSize.

- The attribute ImplementationDataTypeElement.arraySize shall be defined.

- The attribute ImplementationDataTypeElement.arraySizeHandling shall be set to the value allIndicesSameArraySize.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1321] Profile **VSA_FULLY_FLEXIBLE** for **ImplementationDataType** ⌈If the value of attribute ImplementationDataType.dynamicArraySizeProfile is set to the value VSA_FULLY_FLEXIBLE, the ImplementationDataType shall aggregate a VSA Payload ImplementationDataTypeElement that fulfills all the following conditions:

- The attribute ImplementationDataTypeElement.arraySizeSemantics shall not be defined.

- The attribute ImplementationDataTypeElement.category shall be set to the value ARRAY.

- The attribute ImplementationDataTypeElement.arraySize shall not be defined

- The attribute `ImplementationDataTypeElement.arraySizeHandling` shall not be defined.

The `VSA Payload` `ImplementationDataTypeElement` shall immediately aggregate another `ImplementationDataTypeElement` (representing the first dimension) that shall fulfill all the following conditions:

- The attribute `ImplementationDataTypeElement.category` shall be set to STRUCTURE

- The attribute `ImplementationDataTypeElement.arraySizeSemantics` shall be set to the value `variableSize`.

- The attribute `ImplementationDataTypeElement.arraySize` shall be defined.

- The attribute `ImplementationDataTypeElement.arraySizeHandling` shall be set to the value `allIndicesDifferentArraySize`.

The `ImplementationDataTypeElement` shall aggregate another `ImplementationDataTypeElement` that fulfills the following conditions:

- The attribute `ImplementationDataTypeElement.arraySizeSemantics` shall not be defined.

- The attribute `ImplementationDataTypeElement.category` shall be set to the value `ARRAY`.

- The attribute `ImplementationDataTypeElement.arraySize` shall not be defined.

- The attribute `ImplementationDataTypeElement.arraySizeHandling` shall not be defined.

The **aggregation chain is continued** by a (possible empty) sequence of a pair of `ImplementationDataTypeElement`s with the following characteristics:

- The first `ImplementationDataTypeElement` in the pair shall fulfill all the following conditions:

  – The attribute `ImplementationDataTypeElement.category` shall be set to `STRUCTURE`.

  – The attribute `ImplementationDataTypeElement.arraySizeSemantics` shall be set to the value `variableSize`.

  – The attribute `ImplementationDataTypeElement.arraySize` shall be defined.

  – The attribute `ImplementationDataTypeElement.arraySizeHandling` shall be set to the value `allIndicesDifferentArraySize`.

- The second `ImplementationDataTypeElement` in the pair shall fulfill all the following conditions:

- – The attribute `ImplementationDataTypeElement.arraySizeSemantics` shall not be defined.

- – The attribute `ImplementationDataTypeElement.category` shall be set to the value `ARRAY`.

- – The attribute `ImplementationDataTypeElement.arraySize` shall not be defined.

- – The attribute `ImplementationDataTypeElement.arraySizeHandling` shall not be defined.

The **terminating** `ImplementationDataTypeElement` in the aggregation chain shall fulfill all the following conditions:

- The attribute `ImplementationDataTypeElement.arraySizeSemantics` shall be set to the value `variableSize`.

- The attribute `ImplementationDataTypeElement.arraySize` shall be defined.

- The attribute `ImplementationDataTypeElement.arraySizeHandling` shall be set to the value `allIndicesSameArraySize`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1322] `Size Indicator` **for undefined `dynamicArraySizeProfile`** ⌈If the `ImplementationDataType.dynamicArraySizeProfile` does not exist but the `ImplementationDataType` is mapped to an `ApplicationArrayDataType` where the attribute `ApplicationArrayDataType.dynamicArraySizeProfile` exists, then the `ImplementationDataType` shall have the `category STRUCTURE`, representing a `Variable-Size Array Data Type` with `Size Indicator` enabled **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1363] Existence of attributes of `DiagnosticValueNeeds` ⌈if `DiagnosticValueNeeds` is aggregated by a `SwcServiceDependency` in the role `serviceNeeds` then the attributes

- `DiagnosticValueNeeds.diagnosticValueAccess`

- `DiagnosticValueNeeds.dataLength`

shall **not** exist **at any time in the workflow**.

⌋*()*

[constr_1364] Existence of attributes of `DiagnosticIoControlNeeds` ⌈if `DiagnosticIoControlNeeds` is aggregated by a `SwcServiceDependency` in the role `serviceNeeds` then the attributes

- DiagnosticIoControlNeeds.freezeCurrentStateSupported

- DiagnosticIoControlNeeds.shortTermAdjustmentSupported

shall **not** exist **at any time in the workflow**.

⌋*()*

[constr_1375] Existence of attributes of `CompuMethod` and related meta-classes depending on the value of the `category` ⌈

Attributes of `CompuMethod`	IDENTICAL	LINEAR	SCALE_LINEAR	RAT_FUNC	SCALE_RAT_FUNC	TEXTTABLE	BITFIELD_TEXTTABLE	SCALE_LINEAR_AND_TEXTTABLE	SCALE_RATIONAL_AND_TEXTTABLE	TAB_NOINTP
Attribute Existence per Category										
compuInternalToPhys	N/A	D(1)	D(1)	D(2)	D(2)	D	D	D(8)	D(2)	D
compuPhysToInternal	N/A	D(1)	D(1)	D(2)	D(2)	N/A	N/A	N/A	D(2,3)	N/A
Attributes of meta-classes related to CompuMethod										
compuDefaultValue	N/A	O(6)	O(6)	O(6)	O(6)	O(6)	O(6)	O(6)	O(6)	O(6)
CompuScale	N/A	D/1..1	D/1..n	D/1..1	D/1..n	D/1..n	D/1..n	D/1..n	D/1..n	D/1..n
CompuScale.compuInverseValue	N/A	N/A	N/A	O(2)	O(2)	O(5)	N/A	O(2,5)	O(2,5)	O(5)
CompuScale.lowerLimit	N/A	O	D	D(4)	D(4)	D	D	D	D(4)	D
CompuScale.mask	N/A	N/A	N/A	N/A	N/A	N/A	D	N/A	N/A	N/A
CompuScale.shortLabel	N/A	N/A	N/A	N/A	N/A	O(7)	O(7)	O(7)	O(7)	N/A
CompuScale.symbol	N/A	N/A	N/A	N/A	N/A	O(7)	O(7)	O(7)	O(7)	N/A
CompuScale.upperLimit	N/A	O	D	D(4)	D(4)	D	D	D	D(4)	D
CompuConst	N/A	N/A	N/A	N/A	N/A	D/vt	D/vt	D/vt	D/vt	D/vt or vf
CompuRationalCoeffs	N/A	D	D	D	D	N/A	N/A	D	D	N/A
CompuRationalCoeffs.compuDenominator	N/A	D/1v	D/1v	D	D	N/A	N/A	D/1v	D	N/A
CompuRationalCoeffs.compuNumerator	N/A	D/2v	D/2v	D	D	N/A	N/A	D/2v	D	N/A

⌋*()*

[constr_1381] Appearance of core-related possible values of `MemorySection. option` or `SwAddrMethod.option` ⌈Any given collection of values stored in the attributes MemorySection.option or SwAddrMethod.option according to [TPS_-SWCT_01456] shall at most include a single value out of the following list **at the time when the RTE is generated**:

- **coreGlobal**

- **coreLocal**

⌋*()*

[constr_1382] Mutually exclusive existence of attributes `SwVariableRefProxy`. `autosarVariable` vs. `SwVariableRefProxy.mcDataInstanceVar` ⌈In any given AUTOSAR model, the aggregations `SwVariableRefProxy`.`autosarVariable` and `SwVariableRefProxy`.`mcDataInstanceVar` shall never exist at the same time **at any time in the workflow**.

⌋*()*

[constr_1383] Existence of `CompuMethod` and `DataConstr` for `ImplementationDataType`s of `category TYPE_REFERENCE` ⌈The existence of `ImplementationDataType`.`swDataDefProps`.`compuMethod` and `ImplementationDataType`.`swDataDefProps`.`dataConstr` for `ImplementationDataType`s of `category TYPE_REFERENCE` is only allowed, if the respective `ImplementationDataType`, after all type references are resolved, ends up in an `ImplementationDataType` of `category VALUE`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1384] Definition of `invalidValue` for `DataPrototype` typed by `ApplicationPrimitiveDataType` of `category CURVE`, `MAP`, `CUBOID`, `CUBE_4`, `CUBE_5`, `COM_AXIS`, `RES_AXIS`, and `VAL_BLK` ⌈An `invalidValue` **shall not** be specified for a `DataPrototype` typed by `ApplicationPrimitiveDataType` of `category CURVE`, `MAP`, `CUBOID`, `CUBE_4`, `CUBE_5`, `COM_AXIS`, `RES_AXIS`, and `VAL_BLK` **at any time in the workflow**.

⌋*()*

[constr_1385] `DataPrototype` is typed by an `ImplementationDataType` ⌈If a `DataPrototype` is typed by an `ImplementationDataType`, its `initValue` shall **not** be provided by an `ApplicationValueSpecification`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1386] `PortDefinedArgumentValue` shall only be defined for `AbstractProvidedPortPrototype` ⌈A `PortAPIOption` which aggregates at least one `PortDefinedArgumentValue` in the role `portArgValue` shall reference an `AbstractProvidedPortPrototype` typed by a `ClientServerInterface` in the role `port` **at the time when the RTE is generated**.

⌋*()*

[constr_1388] `VariationPointProxy` of `category VALUE` shall not mix "pre-build" and "post-build" use-cases ⌈If the value of `category` of the `VariationPointProxy` is set to `VALUE` then there can only be one value yield from the evaluation of a `VariationPointProxy`. In other words, a `VariationPointProxy` of `category VALUE` shall not mix the "pre-build" and "post-build" use-cases.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1389] Restriction regarding the value of `category` of `VariationPointProxy`.`implementationDataType` ⌈`VariationPointProxy.implementationDataType` shall **not** be of `category` `STRUCTURE`, `ARRAY`, `UNION`, `FUNCTION_REFERENCE`, and `DATA_REFERENCE`.

The `VariationPointProxy.implementationDataType` shall be of `category` `VALUE` or `TYPE_REFERENCE` that, after all references are resolved, yields an `ImplementationDataType` of `category` `VALUE`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1390] Restriction to the value of `SenderReceiverInterface`.`invalidationPolicy`.`handleInvalid` ⌈If the value of `SenderReceiverInterface.invalidationPolicy.handleInvalid` is set to any value other than `HandleInvalidEnum.dontInvalidate` then the `invalidValue` shall not be within the interval defined by the `CompuMethod` of the applicable `dataElement` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1391] Compatibility of `Unit`s in the context of assignment using an `ApplicationValueSpecification` ⌈If an `ApplicationValueSpecification` is used in the context of an assignment to an `AutosarDataPrototype`, then the `ApplicationValueSpecification.swValueCont.unit` shall be compatible to the `Unit` used in the definition of the given `AutosarDataPrototype`, i.e. `AutosarDataType.swDataDefProps.unit`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1392] Compatibility of `Unit`s in the context of assignment using an `ApplicationRuleBasedValueSpecification` ⌈If an `ApplicationRuleBasedValueSpecification` is used in the context of an assignment to an `AutosarDataPrototype` then the `ApplicationRuleBasedValueSpecification.swValueCont.unit` shall be compatible to the `Unit` used in the definition of the given `AutosarDataPrototype`, i.e. `AutosarDataType.swDataDefProps.unit`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1393] Existence of `RuleBasedValueCont.unit` ⌈For every `RuleBased-ValueCont`, the reference `unit` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1395] `NvBlockDataMapping` shall be complete ⌈If an `NvBlockDataMapping` refers to *sub-elements* or *leaf* elements of the `NvDataInterface.nvData` in the context of a particular `PortPrototype`, then **all remaining** *sub-elements* or *leaf* elements **shall effectively be mapped** according to [TPS_SWCT_01659] by means of a collection of `NvBlockDataMapping`s **at the time when the RTE is generated**.

⌋*()*

[constr_1396] Restriction for the value of attribute `category` for non-terminating `ImplementationDataTypeElement`s taken to model a `Variable-Size Array Data Type` ⌈The value of attribute `category` for non-terminating `ImplementationDataTypeElement`s taken to model a `Variable-Size Array Data Type` shall **not** be set to `TYPE_REFERENCE` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1397] Existence of attributes of `TransformerHardErrorEvent` ⌈For any given `TransformerHardErrorEvent`, **either** the attribute `TransformerHardErrorEvent.operation` **or** `TransformerHardErrorEvent.requiredTrigger` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1398] Existence of attributes of `BaseTypeDirectDefinition` ⌈If the value of attribute `BaseTypeDirectDefinition.baseTypeEncoding` is set to `UTF--16` then the attribute `BaseTypeDirectDefinition.byteOrder` shall exist.

The only allowed values of `BaseTypeDirectDefinition.byteOrder` in this case are `mostSignificantByteFirst` and `mostSignificantByteLast`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1399] Standardized values of `ModeDeclarationGroup.category` ⌈The AUTOSAR standard defines the following values of the attribute `ModeDeclarationGroup.category` with a standardized meaning:

- EXPLICIT_ORDER

- ALPHABETIC_ORDER

[TPS_SWCT_01010] defines the meaning of these values.

It is **not allowed** to define any custom or project-specific value of the attribute `ModeDeclarationGroup.category`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1400] Reference to a specific `DataTransformation` ⌈A specific `DataTransformation` shall only be referenced by either

- a `DataPrototypeMapping` in the role `firstToSecondDataTransformation` (and potentially `secondToFirstDataTransformation`) **or**

- an `ISignal` in the role `dataTransformation` **or**

- an `ISignalGroup` in the role `comBasedSignalGroupTransformation` **or**

- a `ClientServerOperationMapping` in the role `firstToSecondDataTransformation`

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_1401] Restrictions on the relation between `DataPrototypeMapping` and `DataTransformation` ⌈A `VariableDataPrototype` in the context of a `PortPrototype` shall − **at the time when the RTE is generated** − **not** be referenced by a `DataPrototypeMapping` that references a `DataTransformation` while a `DataMapping` exists that points to this `VariableDataPrototype` (via the `SystemSignal`) that also refers to an `ISignal` that in turn references a `DataTransformation`.

⌋*()*

[constr_1402] Applicability of core-related possible values of `MemorySection.option` or `SwAddrMethod.option` related to `SwAddrMethod.sectionInitializationPolicy` ⌈If the attribute `SwAddrMethod.option` or `MemorySection.option` is set to `coreLocal` then the attribute `SwAddrMethod.sectionInitializationPolicy` of the same `SwAddrMethod` respectively the `MemorySection.swAddrmethod` shall be either set to `INIT` or `CLEARED` **at the time when the RTE is generated**.

⌋*()*

[constr_1403] `NvBlockDataMapping`s to a given `nvData` shall be unambiguous ⌈If an `NvBlockDataMapping` exists that **directly** and **completely** maps a specific `NvDataInterface.nvData` in the context of a particular `PortPrototype`, then **no** other `NvBlockDataMapping` which maps sub-elements of the `NvDataInterface.nvData` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1404] All NvDataInterface.nvData of PortPrototypes in the context of a specific SwcServiceDependency shall be mapped to the same NvBlock-Descriptor ⌈In the context of a given SwcServiceDependency (which, in turn, is owned by an AtomicSwComponentType), **all** NvDataInterface.nvData of PortPrototypes referenced by a RoleBasedPortAssignment with attribute RoleBasedPortAssignment.role set to NvDataPort shall be connected (either directly or via the definition of suitable PortInterfaceMappings) to NvDataInterface.nvData (on the side of the NvBlockSwComponentType) that are **completely mapped** (via NvBlockDataMappings) **to the identical** NvBlockDescriptor.ramBlock.

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_1407] Definition of SwDataDefProps.dataConstr depending on the capabilities of the data type ⌈The definition of a SwDataDefProps.dataConstr according to [constr_1288] and [constr_1289] is only supported for a DataPrototype of category ARRAY if the corresponding ApplicationArrayDataType or ImplementationDataType of category ARRAY also supports the specification of a SwDataDefProps.dataConstr.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1408] Definition of SwDataDefProps.displayFormat depending on the capabilities of the data type ⌈The definition of a SwDataDefProps.displayFormat according to [constr_1288] and [constr_1289] is only supported for a DataPrototype of category ARRAY if the corresponding ApplicationArrayDataType or ImplementationDataType of category ARRAY also supports the specification of a SwDataDefProps.displayFormat.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1409] Definition of SwDataDefProps.dataConstr depending on the capabilities of the element data type ⌈The definition of a SwDataDefProps.dataConstr according to [constr_1007] and [constr_1009] is only supported for an ApplicationArrayDataType or an ImplementationDataType of category ARRAY if the aggregated ApplicationArrayDataType.element or ImplementationDataType.subElement also supports the specification of a SwDataDefProps.dataConstr.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1410] Definition of `SwDataDefProps.displayFormat` depending on the capabilities of the element data type ⌈The definition of a `SwDataDefProps.displayFormat` according to [constr_1007] and [constr_1009] is only supported for an `ApplicationArrayDataType` or an `ImplementationDataType` of `category ARRAY` if the aggregated `ApplicationArrayDataType.element` or `ImplementationDataType.subElement` also supports the specification of a `SwDataDefProps.displayFormat`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1413] Definition of `SwDataDefProps.stepSize` depending on the capabilities of the data type ⌈The definition of a `SwDataDefProps.stepSize` according to [constr_1288] and [constr_1289] is only supported for a `DataPrototype` of `category ARRAY` if the corresponding `ApplicationArrayDataType` or `ImplementationDataType` of `category ARRAY` also supports the specification of a `SwDataDefProps.stepSize`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1414] Definition of `SwDataDefProps.stepSize` depending on the capabilities of the element data type ⌈The definition of a `SwDataDefProps.stepSize` according to [constr_1007] and [constr_1009] is only supported for an `ApplicationArrayDataType` or an `ImplementationDataType` of `category ARRAY` if the aggregated `ApplicationArrayDataType.element` or `ImplementationDataType.subElement` also supports the specification of a `SwDataDefProps.stepSize`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1415] Supported values of `ModeSwitchEventTriggeredActivity.role` ⌈The only supported value of `ModeSwitchEventTriggeredActivity.role` **at the time when the RTE is generated** is `WriteBlock`.

⌋*()*

[constr_1416] Existence of `ApplicationArrayElement.maxNumberOfElements` ⌈The attribute `ApplicationArrayElement.maxNumberOfElements` shall exist **at the time when the contract phase generation is executed** for all `ApplicationArrayElement`s defined in the scope of an `ApplicationArrayDataType` where the attribute `ApplicationArrayDataType.dynamicArraySizeProfile` does not exist.

⌋*()*

[constr_1417] Invalid connection between `NvBlockSwComponentType` and other `AtomicSwComponentType` (I) ⌈A configuration where an `RPortPrototype` owned by an `AtomicSwComponentType` is simultaneously and directly connected to `AbstractProvidedPortPrototype`s of a collection of `AtomicSwComponentType`s where at least one in the collection is an `NvBlockSwComponentType` for a matching set of `dataElement`s in all these `PortPrototype`s shall be considered invalid **at the time when the RTE is generated**.

⌋*()*

[constr_1418] Invalid connection between `NvBlockSwComponentType` and other `AtomicSwComponentType` (II) ⌈A configuration where a `PRPortPrototype` owned by an `AtomicSwComponentType` is connected to a `PPortPrototype` owned by an `NvBlockSwComponentType` for a matching set of `dataElement`s in all these `PortPrototype`s shall be considered invalid **at the time when the RTE is generated**.

⌋*()*

[constr_1420] Existence of `SwAxisIndividual.inputVariableType` ⌈If the reference `SwAxisIndividual.inputVariableType` does not exist then either:

- `SwAxisIndividual.dataConstr`

- `SwAxisIndividual.unit`

or

- `SwAxisIndividual.dataConstr`

- `SwAxisIndividual.compuMethod.unit`

shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1422] Value of `category` is `VOID` ⌈If the value of the attribute `SwBaseType.category` is set to `VOID` then the attribute `baseTypeSize` and `baseTypeEncoding` shall not exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1423] Completeness of references `ArVariableInImplementationDataInstanceRef.contextDataPrototype` ⌈The reference `ArVariableInImplementationDataInstanceRef.contextDataPrototype` shall be defined for

- each *leaf* (i.e. the end of a chain of aggregating elements) `ImplementationDataTypeElement` of category `TYPE_REFERENCE` in a chain of referencing `ImplementationDataType`s which is not the `targetDataPrototype`

- and each `ImplementationDataTypeElement` owned by an `Implementa-tionDataType` or `ImplementationDataTypeElement` of `category ARRAY` in a chain of referencing `ImplementationDataType`s

starting from the `ImplementationDataType`s of the `rootVariableDataProto-type` down to the leaf `ImplementationDataTypeElement` which is typed (directly or indirectly via `ImplementationDataType` of `category TYPE_REFERENCE`) by the `ImplementationDataType` of the `targetDataPrototype`.

This rule shall be imposed **at the time when the contract phase genera-tion is executed**.

⌋*()*

[constr_1424] Existence of `ArVariableInImplementationDataInstanceRef`. `contextDataPrototype` ⌈The attribute `ArVariableInImplementation-DataInstanceRef.contextDataPrototype` shall only exist **at any time in the workflow** for an `ImplementationDataTypeElement category TYPE_-REFERENCE` or `ARRAY`.

⌋*()*

[constr_1425] Definition of `swCalprmAxisSet.swCalprmAxis` / `SwAxisIndi-vidual.swVariableRef` depending on the capabilities of the data type ⌈The definition of a `swCalprmAxisSet.swCalprmAxis` / `SwAxisIndividual.swVari-ableRef` in the context of an `InstantiationDataDefProps` or a `Parameter-Access` is only supported for a `DataPrototype` of `category ARRAY` if the data type of the `ApplicationArrayElement` also supports the specification of a `swCal-prmAxisSet.swCalprmAxis` / `SwAxisIndividual.swVariableRef` according to [constr_1289].

Thereby, multiple `ApplicationArrayDataType`s might be nested to express mul-tiple array dimensions. This rule shall be imposed **at the time when the con-tract phase generation is executed**.

⌋*()*

[constr_1426] Consistency of array sizes for axes and input variable array ⌈The number of array dimension defined by `ApplicationArrayDataType`s and the val-ues of the `maxNumberOfElements` attributes for the array of elements of `category CURVE`, `MAP`, `CUBOID`, `CUBE_4`, `CUBE_5`, `COM_AXIS`, or `RES_AXIS` shall be **iden-tical** to the number of array dimension and according value of the `maxNumberO-fElements` of the `VariableDataPrototype` referenced by `SwAxisIndividual`. `swVariableRef.autosarVariable`.

This rule shall be imposed **at the time when the contract phase genera-tion is executed**.

⌋*()*

[constr_1427] Definition of `swCalprmAxisSet.swCalprmAxis` / `SwAxisGrouped`. `swCalprmRef` depending on the capabilities of the data type ⌈The definition of a

`swCalprmAxisSet`.`swCalprmAxis`/ `SwAxisGrouped`.`swCalprmRef` in the context of an `InstantiationDataDefProps` or a `ParameterAccess` is only supported for a `DataPrototype` of category ARRAY if the data type of the `Application-ArrayElement` also supports the specification of a `swCalprmAxisSet`.`swCalprmAxis`/ `SwAxisGrouped`.`swCalprmRef` according to [constr_1289].

Thereby, multiple `ApplicationArrayDataType`s might be nested to express multiple array dimensions. This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1428] Consistency of array sizes for arrays of elements of `category CURVE`, `MAP`, `CUBOID`, `CUBE_4`, or `CUBE_5` arrays and used group axes arrays ⌈The number of array dimension defined by `ApplicationArrayDataType`s and the values of attribute `maxNumberOfElements` attributes for the array of elements of `category CURVE`, `MAP`, `CUBOID`, `CUBE_4`, or `CUBE_5` needs to be identical to the number of array dimension and according value of the `maxNumberOfElements` of the `DataPrototype` referenced by `SwAxisGrouped`.`swCalprmRef`.`arParameter` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1429] Access to data within `PortPrototype`s from within `RunnableEntity`s ⌈For a `VariableAccess` that is aggregated in the roles

- `RunnableEntity`.`dataWriteAccess`

- `RunnableEntity`.`dataReadAccess`

- `RunnableEntity`.`dataSendPoint`

- `RunnableEntity`.`dataReceivePointByArgument`

- `RunnableEntity`.`dataReceivePointByValue`

the existence of the following attributes is not allowed:

- `VariableAccess`.`accessedVariable`.`autosarVariable`.`contextDataPrototype`

- `VariableAccess`.`accessedVariable`.`autosarVariable`.`rootVariableDataPrototype`

- `VariableAccess`.`accessedVariable`.`autosarVariableInImplDatatype`

- `VariableAccess`.`accessedVariable`.`localVariable`

In other words: in this case, only the references

- `VariableAccess`.`accessedVariable`.`autosarVariable`.`portPrototype` and

- `VariableAccess.accessedVariable.autosarVariable.targetDataPrototype`

shall exist and the latter shall **exclusively** refer to a `VariableDataPrototype` that is aggregated as either

- `SenderReceiverInterface.dataElement` or

- `NvDataInterface.nvData`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1430] Access to local data from within `RunnableEntity`s ⌈For `VariableAccess` that is aggregated in the roles

- `RunnableEntity.writtenLocalVariable`

- `RunnableEntity.readLocalVariable`

the existence of the following attributes is not allowed:

- `VariableAccess.accessedVariable.autosarVariableInImplDatatype`

- `VariableAccess.accessedVariable.autosarVariable`

In other words, **only** the reference `VariableAccess.accessedVariable.localVariable` shall be used in this case.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1431] Access to parameters from within `RunnableEntity`s ⌈For a `ParameterAccess` that is aggregated in the role `RunnableEntity.parameterAccess` the existence of the following attributes is not allowed:

- `ParameterAccess.accessedParameter.autosarParameter.contextDataPrototype`

- `ParameterAccess.accessedParameter.autosarParameter.rootParameterDataPrototype`

In other words: in this case, **one** of the following alternatives is allowed to exist:

- a combination of

  - `ParameterAccess.accessedParameter.autosarParameter.portPrototype` and

- – ParameterAccess.accessedParameter.autosarParameter.targetDataPrototype that **exclusively** refers to a ParameterDataPrototype aggregated by a ParameterInterface in the role parameter.

  • ParameterAccess.accessedParameter.localParameter that refers to a ParameterDataPrototype that is either aggregated as

    – InternalBehavior.constantMemory or

    – SwcInternalBehavior.perInstanceParameter or

    – SwcInternalBehavior.sharedParameter.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1432] Multiplicity of CommunicationBufferLocking ⌈In a concrete aggregated set of PortAPIOption.supportedFeature, CommunicationBufferLocking shall exist **at most once at the time when the RTE is generated**.

⌋*()*

[constr_1433] Transient faults are not applicable to software-components ⌈An ErrorTracerNeeds aggregated in the context of a SwcInternalBehavior is - **at the time when the RTE is generated** - not allowed to own a TransientFault in the role ErrorTracerNeeds.tracedFailure.

⌋*()*

[constr_1434] CompuScales shall not have identical CompuScale Value Symbolic Names ⌈In a CompuMethod that is subject to [constr_1146], no two CompuScales shall have identical CompuScale Value Symbolic Names (according to [TPS_SWCT_01696]) **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1438] ApplicationArrayElement.indexDataType needs to refer to a CompuMethod of category TEXTTABLE ⌈The reference ApplicationArrayElement.indexDataType shall only point to an ApplicationPrimitiveDataType that in turn refers to a CompuMethod of category TEXTTABLE.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1439] Requirements on ApplicationArrayElement if attribute indexDataType exists ⌈If ApplicationArrayElement.indexDataType exists then the attribute ApplicationArrayElement.arraySizeSemantics shall be set to the

value `fixedSize` and attribute `arraySizeHandling` shall not exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1440] Size of the `CompuMethod` of `category` `TEXTTABLE` referenced by `ApplicationArrayElement.indexDataType` ⌈The interval defined by the `CompuScale`s contained in the `CompuMethod` referenced by `ApplicationArrayElement.indexDataType` shall start at 0 and include all integer values until `ApplicationArrayElement.maxNumberOfElements` - 1.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1442] `category TYPE_REFERENCE` shall not be used for modeling the "payload" of a `Wrapped Union Data Type` ⌈For the modeling of the "payload" part of a `Wrapped Union Data Type` it shall not be possible **at any time in the workflow** to use an `ImplementationDataTypeElement` of `category TYPE_-REFERENCE` that finally (i.e. after all possible indirections are resolved) boils down to `category UNION`.

⌋*()*

[constr_1444] Limited applicability of `Wrapped Union Data Type` ⌈There is no support **at any time in the workflow** for the usage of `Wrapped Union Data Type` in `PortInterfaceMapping`s, and Diagnostics.

⌋*()*

[constr_1445] Initialization of the `Member Selector` of a `Wrapped Union Data Type` ⌈The `initValue` for the `Member Selector` shall **never be set to any value other than 1**.

This rule shall be imposed **at any time in the workflow**.

⌋*()*

[constr_1446] No definition of `invalidValue` for a `Wrapped Union Data Type` ⌈The definition of an `invalidValue` for a `DataPrototype` typed by a `Wrapped Union Data Type` is not supported **at any time in the workflow**.

⌋*()*

[constr_1468] Limitation on the number of `SwcExclusiveAreaPolicy`s ⌈An `ExclusiveArea` shall only be referenced by **at most** one `SwcExclusiveAreaPolicy` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1469] Applicability of constraints depending on the existence of a data transformation ⌈[constr_1269], [constr_1270], [constr_1268], and [constr_1240] shall — **at any time in the workflow** — **not** apply under the following conditions:

- A reference from the respective `ClientServerOperationMapping` to a `DataTransformation` in the role `firstToSecondDataTransformation` exists.

- The value of the attribute `dataTransformationKind` of the referenced `DataTransformation` is set to `DataTransformationKindEnum.asymmetricFromByteArray` or `DataTransformationKindEnum.asymmetricToByteArray`.

⌋*()*

[constr_1516] Completeness of references `ArParameterInImplementationDataInstanceRef.contextDataPrototype` ⌈The reference `ArParameterInImplementationDataInstanceRef.contextDataPrototype` shall be defined for

- each *leaf* (i.e. the end of a chain of aggregating elements) `ImplementationDataTypeElement` of category `TYPE_REFERENCE` in a chain of referencing `ImplementationDataType`s which is not the `targetDataPrototype`

- and each `ImplementationDataTypeElement` owned by an `ImplementationDataType` or `ImplementationDataTypeElement` of category `ARRAY` in a chain of referencing `ImplementationDataType`s

starting from the `ImplementationDataType`s of the `rootParameterDataPrototype` down to the leaf `ImplementationDataTypeElement` which is typed (directly or indirectly via `ImplementationDataType` of category `TYPE_REFERENCE`) by the `ImplementationDataType` of the `targetDataPrototype`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1517] Existence of `ArParameterInImplementationDataInstanceRef.contextDataPrototype` ⌈The attribute `ArParameterInImplementationDataInstanceRef.contextDataPrototype` shall only exist **at any time in the workflow** for an `ImplementationDataTypeElement` category `TYPE_REFERENCE` or `ARRAY`.

⌋*()*

[constr_1518] Consistency of data types in the context of `ArParameterInImplementationDataInstanceRef` ⌈The definition of attributes `contextDataPrototype` and `targetDataPrototype` shall be enclosed in the context of the definition of the data type used to type `rootParameterDataPrototype` **at any time in the workflow**.

⌋*()*

[constr_1519] Existence of attributes vs. `category` of `ApplicationValueSpecification` ⌈

Attribute of `ApplicationValueSpecification`	Attribute Existence per Category									
	VALUE	STRING	BOOLEAN	COM_AXIS	RES_AXIS	CURVE	MAP	CUBOID	CUBE_4	CUBE_5
`swValueCont`	D	D	D	D	D	D	D	D	D	D
`swValueCont.unit`	O	O	O	O	O	O	O	O	O	O
`swValueCont.swValuesPhys`	D	D	D	D	D	D	D	D	D	D
`swValueCont.swArraysize`	N/A	N/A	N/A	D	D	D	D	D	D	D
`swAxisCont`	N/A	N/A	N/A	N/A	D	D	D	D	D	D
`swAxisCont.unit`	N/A	N/A	N/A	N/A	O	O	O	O	O	O
`swAxisCont.category`	N/A	N/A	N/A	N/A	D	D	D	D	D	D
`swAxisCont.swAxisIndex`	N/A	N/A	N/A	N/A	D	D	D	D	D	D
`swAxisCont.swArraysize`	N/A	N/A	N/A	N/A	D	D	D	D	D	D
`swAxisCont.swValuesPhys`	N/A	N/A	N/A	N/A	D	O(1)	O(1)	O(1)	O(1)	O(1)

⌋*()*

[constr_1520] Semantics of `ObdRatioServiceNeeds.rateBasedMonitoredEvent` ⌈In the context of an `SwcServiceDependency`, each `DiagnosticEventNeeds` referenced in the role `rateBasedMonitoredEvent` shall only be referenced by at most a single `ObdRatioServiceNeeds` **at the time when the RTE is generated**.

⌋*()*

[constr_1521] Reference from `AsynchronousServerCallReturnsEvent` to `AsynchronousServerCallResultPoint` ⌈In the context of a `RunnableEntity`, a given `AsynchronousServerCallResultPoint` shall only be referenced by one `AsynchronousServerCallReturnsEvent` in the role `eventSource` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1523] No mode disabling for `OperationInvokedEvent`s ⌈An `OperationInvokedEvent` shall not have a reference to a `ModeDeclaration` in the role `disabledMode` **at the time when the RTE is generated**.

⌋*()*

[constr_1538] Restriction for `ReceiverComSpec.dataElement` ⌈The reference `ReceiverComSpec.dataElement` **shall not** refer to an `ArgumentDataPrototype` or `ParameterDataPrototype` **at any time in the workflow**.

⌋*()*

[constr_1539] Restriction for `SenderComSpec.dataElement` ⌈The reference `SenderComSpec.dataElement` **shall not** refer to an `ArgumentDataPrototype` or

`ParameterDataPrototype` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1540] Existence of `ClientComSpec.operation` ⌈The reference `ClientComSpec.operation` **shall exist** if the `AbstractRequiredPortPrototype` that owns the `ClientComSpec` is typed by a `ClientServerInterface`. This rule shall be imposed **at any time in the workflow**.

⌋*()*

[constr_1541] Existence of `ServerComSpec.operation` ⌈The reference `ServerComSpec.operation` **shall exist** if the `AbstractProvidedPortPrototype` that owns the `ServerComSpec` is typed by a `ClientServerInterface`.

This rule shall be imposed **at any time in the workflow**.

⌋*()*

[constr_1544] Standardized values and multiplicities for the modeling of `SwAxisGeneric` for the definition of a fix axis ⌈

category of swAxisType	category of SwGenericAxisParamType	Multiplicity of swGenericAxisParam	Multiplicity of vf
FIX_AXIS_PAR	OFFSET	1	1
	SHIFT	1	1
FIX_AXIS_PAR_DIST	OFFSET	1	1
	DISTANCE	1	1
FIX_AXIS_PAR_LIST	LIST	1	1..*

⌋*()*

[constr_1545] No initialization for fix axis ⌈An `ApplicationValueSpecification` taken to initialize an `ApplicationPrimitiveDataType` that contains a fix axis shall not contain initial values for the axis index of the fix axis inside the `ApplicationPrimitiveDataType` **at any time in the workflow**.

⌋*()*

[constr_1583] `PortInterfaceMapping` for `DataPrototype` typed by `Compound Primitive Data Type` ⌈There is one very limited use case to apply `PortInterfaceMapping` for a `DataPrototype` typed by a `Compound Primitive Data Type`: adjustment of the `shortName` of the `DataPrototype`. Everything else is **not supported**. This rule shall be imposed **at any time in the workflow**.

⌋*()*

[constr_1592] Definition of `SwDataDefProps.displayPresentation` depending on the capabilities of the data type ⌈The definition of a `SwDataDefProps.displayPresentation` according to [constr_1288] and [constr_1289] shall only be applied for a `DataPrototype` of category ARRAY if the corresponding `Application-`

`ArrayDataType` or `ImplementationDataType` of `category ARRAY` supports the specification of a `SwDataDefProps`.`displayPresentation`.

This rule shall be imposed **at any time in the workflow**.

⌋*()*

[constr_1602] Definition of `SwDataDefProps.displayPresentation` depending on the capabilities of the element ⌈The definition of a `SwDataDefProps.displayPresentation` according to [constr_1007] and [constr_1009] is only supported for an `ApplicationArrayDataType` or an `ImplementationDataType` of `category ARRAY` if the aggregated `ApplicationArrayDataType`.element or `ImplementationDataType`.subElement also supports the specification of a `SwDataDefProps.displayPresentation`.

This rule shall be imposed **at any time in the workflow**.

⌋*()*

[constr_1607] Only `Wrapped Union Data Type`s in `PortInterface` ⌈Within the scope of a `PortInterface` the usage of a Union data type is only supported

- for `Wrapped Union Data Type`s.

- for a `PortInterface` that is used to type a `PortPrototype` that does not appear as a context in an `instanceRef` owned by a `DataMapping`. See also [constr_1441].

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1608] Existence of `rootParameterDataPrototype` ⌈The reference `rootParameterDataPrototype` shall exist **at any time in the workflow** if and only if

- `AutosarDataType` of the `autosarParameter` is a composite data type and

- `targetDataPrototype` refers to a `DataPrototype` inside the `rootParameterDataPrototype`.

⌋*()*

[constr_1609] Existence of `rootVariableDataPrototype` ⌈The reference `rootVariableDataPrototype` shall exist **at any time in the workflow** if and only if

- the `AutosarDataType` of the `autosarVariable` is a composite data type and

- the `targetDataPrototype` refers to a `DataPrototype` inside the `rootVariableDataPrototype`.

⌋*()*

[constr_1610] **Existence of `SwDataDefProps.swValueBlockSize` and `Sw-DataDefProps.swValueBlockSizeMult`** ⌈Attributes `SwDataDefProps.swValueBlockSize` and `SwDataDefProps.swValueBlockSizeMult` shall not exist at the same time in the context of a given `SwDataDefProps`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1611] **Existence of `ImplementationDataTypeSubElementRef.implementationDataTypeElement` as opposed to `ImplementationDataTypeSubElementRef.parameterImplementationDataTypeElement`** ⌈For any given `ImplementationDataTypeSubElementRef`, either the aggregation

- `ImplementationDataTypeSubElementRef.implementationDataTypeElement` or

- `ImplementationDataTypeSubElementRef.parameterImplementationDataTypeElement`

shall exist **at any time in the workflow**.

⌋*()*

[constr_1622] **Value of `TimingEvent.offset` vs. `TimingEvent.period`** ⌈If a value is defined for attribute `TimingEvent.offset` then this value shall be greater than 0 and less or equal than the value of attribute `TimingEvent.period` of the respective `TimingEvent` **at the time when the RTE is generated**.

⌋*()*

[constr_1631] **Applicability of `DataPrototypeMapping.secondToFirstDataTransformation`** ⌈The reference to `DataTransformation` in the role `DataPrototypeMapping.secondToFirstDataTransformation` shall only exist if reference `DataPrototypeMapping.firstToSecondDataTransformation` exists and refers to a `DataTransformation` where attribute `dataTransformationKind` exists and is **not** set to the value `symmetric`.

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_1632] **Restriction for `firstToSecondDataTransformation` and `secondToFirstDataTransformation`** ⌈If – **at the time when the RTE is generated** – both the reference `firstToSecondDataTransformation` and the reference `secondToFirstDataTransformation` exist in the context of the same `DataPrototypeMapping` then

- the `firstToSecondDataTransformation` shall refer to a `DataTransformation` with attribute `dataTransformationKind` set to `asymmetricToByteArray` and

- the `secondToFirstDataTransformation` shall refer to a `DataTransformation` with attribute `dataTransformationKind` set to `asymmetricFromByteArray`.

⌋*()*

[constr_1634] Allowed combinations of `ApplicationDataType.category` vs. `CompuMethod.category` ⌈

	IDENTICAL	LINEAR	SCALE_LINEAR	SCALE_LINEAR_AND_TEXTTABLE	RAT_FUNC	SCALE_RATIONAL_AND_TEXTTABLE	TEXTTABLE	TAB_NOINTP	BITFIELD_TEXTTABLE
VALUE	x	x	x	x	x	x	x	x	x
VAL_BLK	x	x	x	x	x	x	x	x	x
BOOLEAN	n/a	n/a	n/a	n/a	n/a	n/a	x	n/a	n/a
CURVE	x	x	x	x	x	x	x	x	x
MAP	x	x	x	x	x	x	x	x	x
CUBOID	x	x	x	x	x	x	x	x	x
CUBE_4	x	x	x	x	x	x	x	x	x
CUBE_5	x	x	x	x	x	x	x	x	x

⌋*()*

[constr_1635] Relevance of attribute `isOptional` ⌈If a `SubElementMapping` is defined for the elements of a structured data type then the attribute `isOptional`[16] shall either not exist for the `firstElement` and `secondElement` or it shall have the identical value for the `firstElement` and `secondElement`. This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_1636] Mapping of data types that represent an `Optional Element Structure` ⌈An `ApplicationRecordDataType` with at least one `element` where attribute `isOptional` is set to `true` shall only be mapped to an `ImplementationDataType` that fulfills the structural requirements to represent an `Optional`

---

[16]this is valid for both `ApplicationRecordElement` and `ImplementationDataTypeElement`

`Element Structure` (see [TPS_SWCT_01774]) **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1637] **Existence of `ImplementationDataTypeElement.isOptional` vs. `ImplementationDataType.isStructWithOptionalElement`** ⌈If one `ImplementationDataType.subElement` sets attribute `isOptional` to the value `true` then the enclosing `ImplementationDataType` shall also set attribute `isStructWithOptionalElement` to `true`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1638] First `ImplementationDataTypeElement` of `ImplementationDataType` that represents an `Optional Element Structure` ⌈The first `ImplementationDataTypeElement` of `ImplementationDataType` that represents an `Optional Element Structure`, i.e. the `availabilityBitfield` according to [TPS_SWCT_01774], shall not set attribute `isOptional` to `true` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1639] `ImplementationDataTypeElement` with attribute `isOptional` set to `true` ⌈`ImplementationDataTypeElement` with attribute `isOptional` set to `true` shall **not** be of `category STRUCTURE` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1640] No use of `Optional Element Structure` for interaction with the diagnostic stack ⌈An `SwcServiceDependency` that aggregates a diagnostic-related subclass of `ServiceNeeds` shall not refer to any `PortPrototype` by means of either a `RoleBasedPortAssignment` or `RoleBasedDataAssignment` where the respective `PortInterface` contains any `DataPrototype` typed by an `Optional Element Structure`.

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_1662] Compatibility of `ApplicationRecordDataType` and `ImplementationDataType` that both represent an `Optional Element Structure` ⌈An `ApplicationRecordDataType` that represents an `Optional Element Structure` shall (after all indirections created by `ImplementationDataType`s of `category TYPE_REFERENCE` are resolved) only be mapped/connected to an `ImplementationDataType` of `category STRUCTURE` that represents an `Optional Element Structure` if corresponding pairs of elements have the same value of the attribute `isOptional`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1679] **Existence of attribute `RoleBasedDataAssignment.usedDataElement.localVariable` for `RoleBasedDataAssignment.role = signalBasedDiagnostics`** ⌈If the attribute `RoleBasedDataAssignment.role` is set to the value `signalBasedDiagnostics` then the reference `RoleBasedDataAssignment.usedDataElement.localVariable` shall not exist **at the time when the RTE is generated**.

⌋*()*

[constr_1680] **Existence of attribute `RoleBasedDataAssignment.usedDataElement.localVariable` for `RoleBasedDataAssignment.role = AppModeRequestInterface`** ⌈If the attribute `RoleBasedDataAssignment.role` is set to the value `AppModeRequestInterface`, then the reference `RoleBasedDataAssignment.usedDataElement.localVariable` shall not exist **at the time when the RTE is generated**.

⌋*()*

[constr_1681] **Existence of attribute `RoleBasedDataAssignment.usedDataElement.localVariable` for `RoleBasedDataAssignment.role = VerificationStatus`** ⌈If the attribute `RoleBasedDataAssignment.role` is set to the value `VerificationStatus` then the reference `RoleBasedDataAssignment.usedDataElement.localVariable` shall not exist **at the time when the RTE is generated**.

⌋*()*

[constr_1682] **Existence of attribute `RoleBasedDataAssignment.usedDataElement.localVariable` for `RoleBasedDataAssignment.role = V2xFacVdp`** ⌈If the attribute `RoleBasedDataAssignment.role` is set to the value `V2xFacVdp` then the reference `RoleBasedDataAssignment.usedDataElement.localVariable` shall not exist **at the time when the RTE is generated**.

⌋*()*

[constr_1694] Allowed target of `SwDataDefProps.implementationDataType` ⌈The reference `SwDataDefProps.implementationDataType` shall only refer to an `ImplementationDataType`. Any other subclass of `AbstractImplementationDataType` is not supported as a reference target.

This rule shall be applied **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1712] **Existence of attribute `ArrayValueSpecification.intendedPartialInitializationCount`** ⌈An `ArrayValueSpecification` where at-

tribute `intendedPartialInitializationCount` exists shall only be applied for the initialization of an `ApplicationArrayDataType` where attribute `arraySizeSemantics` is set to `variableSize`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1713] **NvBlockDescriptor.writingStrategy.usedDataElement shall refer to AutosarDataPrototype** ⌈The reference `NvBlockDescriptor.writingStrategy.usedDataElement` shall **only** refer to an `AutosarDataPrototype` **at the time when the RTE is generated**.

⌋*()*

[constr_1714] **AutosarDataPrototype shall only be referenced by a single NvBlockDescriptor.writingStrategy** ⌈If an `AutosarDataPrototype` in the context of a `PortPrototype` is referenced from a `NvBlockDescriptor.writingStrategy` then this `AutosarDataPrototype` shall not be referenced from any other `NvBlockDescriptor.writingStrategy`.

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_1715] **Possible values of attribute NvBlockDescriptor.writingStrategy.role** ⌈The attribute `NvBlockDescriptor.writingStrategy.role` shall only have one of the following values **at the time when the RTE is generated** (see [TPS_SWCT_01586]):

- **storeAtShutdown**

- **storeImmediate**

- **storeOnChange**

⌋*()*

[constr_1716] **Consistency of attribute NvBlockDescriptor.writingStrategy.role set to storeAtShutdown** ⌈The existence of `NvBlockDescriptor.writingStrategy` where attribute `role` is set to `storeAtShutdown` is only supported if `NvBlockDescriptor.nvBlockNeeds.storeAtShutdown` exists and is set to `true` **at the time when the RTE is generated**.

⌋*()*

[constr_1717] **Consistency of attribute NvBlockDescriptor.writingStrategy.role set to storeImmediate** ⌈The existence of `NvBlockDescriptor.writingStrategy` where attribute `role` is set to `storeImmediate` is only supported if

NvBlockDescriptor.nvBlockNeeds.storeImmediate exists and is set to true **at the time when the RTE is generated**.

⌋*()*

[constr_1718] Inheritance of SwDataDefProps.dataConstr from an array data type to the array elements ⌈A SwDataDefProps.dataConstr specified for an ApplicationArrayDataType or ImplementationDataType of category ARRAY applies to all array leaf elements represented by (potentially multiple levels of) ApplicationArrayDataType.element or ImplementationDataType.subElement.

In this case, the ApplicationArrayDataType.element or ImplementationDataType.subElement shall not have an own SwDataDefProps.dataConstr. This also applies for multi-dimensional array data types.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1719] Inheritance of SwDataDefProps.displayFormat from an array data type to the array elements ⌈A SwDataDefProps.displayFormat specified for an ApplicationArrayDataType or ImplementationDataType of category ARRAY applies to all array leaf elements represented by (potentially multiple levels of) ApplicationArrayDataType.element or ImplementationDataType.subElement.

In this case, the ApplicationArrayDataType.element or ImplementationDataType.subElement shall not have an own SwDataDefProps.displayFormat. This also applies for multi-dimensional array data types.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1720] Inheritance of SwDataDefProps.stepSize from an array data type to the array elements ⌈A SwDataDefProps.stepSize specified for an ApplicationArrayDataType or ImplementationDataType of category ARRAY applies to all array leaf elements represented by (potentially multiple levels of) ApplicationArrayDataType.element or ImplementationDataType.subElement.

In this case, the ApplicationArrayDataType.element or ImplementationDataType.subElement shall not have an own SwDataDefProps.stepSize. This also applies for multi-dimensional array data types.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1724] Usage of attribute `ClientServerOperation.diagArgIntegrity` ⌈With the exception of the context of a `ServiceSwComponentType`, the attribute `ClientServerOperation.diagArgIntegrity` shall only have the value `true` if the `ClientServerInterface` containing the respective `ClientServerOperation` is used to type a `PPortPrototype` that is referenced by a `RoleBasedPortAssignment` aggregated by a `SwcServiceDependency` that in turn aggregates `DiagnosticRoutineNeeds`.

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_1726] Ordering of `MetaDataItemSet.metaDataItem` ⌈The ordering of the elements of `MetaDataItemSet.metaDataItem` shall be **descending** with respect to the value of `MetaDataItem.length`, such that the `MetaDataItem` with the largest value of attribute `length` is located in the first position and the `MetaDataItem` with the smallest value of attribute `length` is located in the last position.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1735] Limitation of the aggregation of `AutosarVariableRef` in the context of an `NvBlockDataMapping` owned by a `BulkNvDataDescriptor` ⌈Any `NvBlockDataMapping` owned by a `BulkNvDataDescriptor` shall only aggregate an `AutosarVariableRef` in the role `readNvData` and `nvRamBlockElement` (that in turn refers to the `BulkNvDataDescriptor.bulkNvBlock`) **at the time when the RTE is generated**.

⌋*()*

[constr_1741] Restriction to explicit sending semantics for the usage of `Data Services` in the context of a `SwcServiceDependency` that aggregates `DiagnosticValueNeeds` that in turn is referenced by a `DiagnosticIoControlNeeds` ⌈A `dataElement`

- that is referenced by a `RoleBasedDataAssignment` (where the attribute `role` is set to `signalBasedDiagnostics`) owned by a `SwcServiceDependency` that aggregates `DiagnosticValueNeeds` that in turn is referenced by a `DiagnosticIoControlNeeds`

- **shall also be referenced** by a `VariableAccess` aggregated in the role `dataSendPoint` by a given `RunnableEntity` that in turn belongs to the enclosing `SwcInternalBehavior`.

- **shall not be referenced** by a `VariableAccess` aggregated in the role `dataWriteAccess` by a given `RunnableEntity` that in turn belongs to the enclosing `SwcInternalBehavior`.

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_1754] **Aggregation of `NumericalRuleBasedValueSpecification`** ⌈Each `ArrayValueSpecification` shall only aggregate at most one `Numerical-RuleBasedValueSpecification` in the role element.

If one `NumericalRuleBasedValueSpecification` is aggregated then it shall be the only aggregated element, i.e. no further `ValueSpecification` shall exist in the same aggregation where an `NumericalRuleBasedValueSpecification` is aggregated.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1755] **Aggregation of `CompositeRuleBasedValueSpecification`** ⌈Each `ArrayValueSpecification` shall only aggregate at most one `CompositeRuleBasedValueSpecification` in the role element.

If one `CompositeRuleBasedValueSpecification` is aggregated then it shall be the only aggregated element, i.e. no further `ValueSpecification` shall exist in the same aggregation where an `CompositeRuleBasedValueSpecification` is aggregated.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1771] Existence of `SwValueCont.unit` ⌈For every `SwValueCont`, the reference `unit` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1773] Value of attribute `dataSendPoint.returnValueProvision` ⌈All `RunnableEntity.dataSendPoint` that refer to the same `accessedVariable` shall define the identical value for attribute `returnValueProvision` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1774] Value of attribute `dataReceivePointByArgument.returnValueProvision` ⌈All `RunnableEntity.dataReceivePointByArgument` that refer to the same `accessedVariable` shall define the identical value for attribute `returnValueProvision` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1775] Value of attribute `serverCallPoint.returnValueProvision` ⌈All `RunnableEntity.serverCallPoint` that refer to the same `operation` shall define the identical value of attribute `returnValueProvision` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1776] Value of attribute `asynchronousServerCallResultPoint.returnValueProvision` ⌈All `RunnableEntity.asynchronousServerCallResultPoint` that refer to the same `AsynchronousServerCallPoint.operation` shall define the identical value of attribute `returnValueProvision` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1777] Value of attribute `externalTriggeringPoint.returnValueProvision` ⌈All `RunnableEntity.externalTriggeringPoint` that refer to the same `trigger` shall define the identical value of attribute `returnValueProvision` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1778] Value of attribute `modeSwitchPoint.returnValueProvision` ⌈All `RunnableEntity.modeSwitchPoint` that refer to the same `modeGroup` shall define the identical value of attribute `returnValueProvision` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1779] Scope of the definition of an `AbstractRuleBasedValueSpecification` ⌈An `AbstractRuleBasedValueSpecification` shall only be defined in the context of an `ArrayValueSpecification` or a `ConstantSpecification`. If the `AbstractRuleBasedValueSpecification` is defined in the context of a `ConstantSpecification` then a reference to this `ConstantSpecification` shall only be created in the context of an `ArrayValueSpecification`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1783] Existence of attribute `ImplementationDataTypeElement.arrayImplPolicy` ⌈Attribute `ImplementationDataTypeElement.arrayImplPolicy` shall only exist **at the time when the contract phase generation is executed** if the enclosing `ImplementationDataType` or `ImplementationDataTypeElement` is of `category ARRAY`.

⌋*()*

[constr_1860] Multiplicity of `DelegationSwConnector.innerPort` ⌈For each `DelegationSwConnector`, the reference `DelegationSwConnector.innerPort`

shall exist **at the time when the creation of the CompositionSwComponentType is finished**.

⌋*()*

[constr_1861] Multiplicity of `DelegationSwConnector.outerPort` ⌈For each `DelegationSwConnector`, the reference `DelegationSwConnector.outerPort` shall exist **at the time when the creation of the CompositionSwComponentType is finished**.

⌋*()*

[constr_1862] Multiplicity of `PassThroughSwConnector.requiredOuterPort` ⌈For each `PassThroughSwConnector`, the reference `PassThroughSwConnector.requiredOuterPort` shall exist **at the time when the creation of the CompositionSwComponentType is finished**.

⌋*()*

[constr_1863] Multiplicity of `PassThroughSwConnector.providedOuterPort` ⌈For each `PassThroughSwConnector`, the reference `PassThroughSwConnector.providedOuterPort` shall exist **at the time when the creation of the CompositionSwComponentType is finished**.

⌋*()*

[constr_1864] Multiplicity of `InstantiationRTEEventProps.refinedEvent` ⌈For each `InstantiationRTEEventProps`, the instance-reference `InstantiationRTEEventProps.refinedEvent` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1865] Existence of `InvalidationPolicy.dataElement` ⌈For each `InvalidationPolicy`, the reference `InvalidationPolicy.dataElement` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1866] Existence of `MetaDataItem.length` ⌈For each `MetaDataItem`, attribute `length` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1867] Existence of `MetaDataItem.metaDataItemType` ⌈For each `MetaDataItem`, attribute `metaDataItemType` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1868] Existence of `MetaDataItemSet.dataElement` ⌈For each `MetaDataItemSet` that aggregates at least one `metaDataItem`, at least one reference to

a `dataElement` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1869] Existence of attribute `ArgumentDataPrototype.direction` ⌈For each `ArgumentDataPrototype`, attribute `direction` shall be defined **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1871] Existence of attribute `ModeRequestTypeMap.implementationDataType` ⌈For each `ModeRequestTypeMap`, attribute `implementationDataType` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1872] Existence of attribute `ModeRequestTypeMap.modeGroup` ⌈For each `ModeRequestTypeMap`, attribute `modeGroup` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1873] Existence of `DataPrototypeMapping.firstDataPrototype` ⌈For each `DataPrototypeMapping`, the reference in the role `firstDataPrototype` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1874] Existence of `DataPrototypeMapping.secondDataPrototype` ⌈For each `DataPrototypeMapping`, the reference in the role `secondDataPrototype` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1875] Existence of reference `ClientServerOperationMapping.firstOperation` ⌈For each `ClientServerOperationMapping`, the reference in the role `firstOperation` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1876] Existence of reference `ClientServerOperationMapping.secondOperation` ⌈For each `ClientServerOperationMapping`, the reference in the role `secondOperation` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1877] Existence of reference `ModeDeclarationGroupPrototypeMapping.firstModeGroup` ⌈For each `ModeDeclarationGroupPrototypeMapping`,

the reference in the role `firstModeGroup` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1878] Existence of reference `ModeDeclarationGroupPrototypeMapping.secondModeGroup` ⌈For each `ModeDeclarationGroupPrototypeMapping`, the reference in the role `secondModeGroup` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1879] Existence of reference `ModeDeclarationMapping.firstMode` ⌈For each `ModeDeclarationMapping`, at least one reference `firstMode` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1880] Existence of reference `ModeDeclarationMapping.secondMode` ⌈For each `ModeDeclarationMapping`, the reference `secondMode` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1881] Existence of reference `TriggerMapping.firstTrigger` ⌈For each `TriggerMapping`, the reference `firstTrigger` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1882] Existence of reference `TriggerMapping.secondTrigger` ⌈For each `TriggerMapping`, the reference `secondTrigger` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1883] Existence of `ApplicationCompositeDataTypeSubElementRef.applicationCompositeElement` ⌈For each `ApplicationCompositeDataTypeSubElementRef`, the reference `applicationCompositeElement` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1884] Existence of attribute `TextTableMapping.identicalMapping` ⌈For each `TextTableMapping`, the attribute `identicalMapping` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1885] Existence of attribute `TextTableMapping.mappingDirection` ⌈For each `TextTableMapping`, the attribute `mappingDirection` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1886] Existence of attribute `TextTableValuePair.firstValue` ⌈For each `TextTableValuePair`, the attribute `firstValue` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1887] Existence of attribute `TextTableValuePair.secondValue` ⌈For each `TextTableValuePair`, the attribute `secondValue` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1888] Existence of attribute `DataTransformation.executeDespite-DataUnavailability` ⌈For each `DataTransformation`, the attribute `executeDespiteDataUnavailability` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1889] Existence of attribute `QueuedReceiverComSpec.queueLength` ⌈For each `QueuedReceiverComSpec`, attribute `queueLength` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1890] Existence of attribute `DataFilter.dataFilterType` ⌈For each `DataFilter`, attribute `dataFilterType` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1891] Existence of attribute `NonqueuedReceiverComSpec.initValue` ⌈For each `NonqueuedReceiverComSpec`, attribute `initValue` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1892] Existence of attribute `TransmissionAcknowledgementRequest.timeout` ⌈For each `TransmissionAcknowledgementRequest`, attribute `timeout` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1893] Existence of attribute `ServerComSpec.queueLength` ⌈For each `ServerComSpec`, attribute `queueLength` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1894] Existence of attribute `ModeSwitchSenderComSpec.queueLength`
⌈For each `ModeSwitchSenderComSpec`, attribute `queueLength` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1895] Existence of attribute `ModeSwitchSenderComSpec.modeGroup`
⌈For each `ModeSwitchSenderComSpec`, attribute `modeGroup` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1896] Existence of attribute `ModeSwitchReceiverComSpec.modeGroup`
⌈For each `ModeSwitchReceiverComSpec`, attribute `modeGroup` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1897] Existence of reference `ParameterProvideComSpec.parameter`
⌈For each `ParameterProvideComSpec`, the reference `parameter` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1898] Existence of reference `ParameterRequireComSpec.parameter`
⌈For each `ParameterRequireComSpec`, the reference `parameter` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1899] Existence of reference `NvRequireComSpec.variable` ⌈For each `NvRequireComSpec`, the reference `variable` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1900] Existence of reference `NvProvideComSpec.variable` ⌈For each `NvProvideComSpec`, the reference `variable` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1901] Existence of attribute `EndToEndDescription.category` ⌈For each `EndToEndDescription`, attribute `category` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1902] Existence of attribute `EndToEndProtection.endToEndProfile`
⌈For each `EndToEndProtection`, attribute `endToEndProfile` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1903] Existence of reference `DataTypeMap.applicationDataType` ⌈For each `DataTypeMap`, reference `applicationDataType` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1904] Existence of reference `DataTypeMap.implementationDataType` ⌈For each `DataTypeMap`, reference `implementationDataType` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1905] Existence of attribute `SwTextProps.arraySizeSemantics` ⌈For each `SwTextProps`, attribute `arraySizeSemantics` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1906] Existence of attribute `SwTextProps.swMaxTextSize` ⌈For each `SwTextProps`, attribute `swMaxTextSize` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1907] Existence of attribute `ApplicationArrayDataType.element` ⌈For each `ApplicationArrayDataType`, the aggregation of `ApplicationArrayElement` in the role `element` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1908] Existence of attribute `ApplicationRecordDataType.element` ⌈For each `ApplicationRecordDataType`, the aggregation of `ApplicationRecordElement` in the role `element` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1909] Existence of attribute `ImplementationProps.symbol` ⌈For each `ImplementationProps`, the attribute `symbol` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1910] Existence of attribute `BaseType.baseTypeDefinition` ⌈For each `BaseType` (which will be utilized in the form of `SwBaseType`), the aggregation in the role `baseTypeDefinition` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1911] Existence of `ArVariableInImplementationDataInstanceRef.targetDataPrototype` ⌈For each `ArVariableInImplementationDataIn-`

stanceRef, the reference `targetDataPrototype` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1912] Existence of reference `ArParameterInImplementationDataInstanceRef.targetDataPrototype` ⌈For each `ArParameterInImplementationDataInstanceRef`, the reference `targetDataPrototype` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1913] Existence of attribute `CompuRationalCoeffs.compuDenominator` ⌈For each `CompuRationalCoeffs`, the attribute `compuDenominator` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1914] Existence of attribute `CompuRationalCoeffs.compuNumerator` ⌈For each `CompuRationalCoeffs`, the attribute `compuNumerator` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1915] Existence of attribute `PhysicalDimensionMapping.firstPhysicalDimension` ⌈For each `PhysicalDimensionMapping`, attribute `firstPhysicalDimension` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1916] Existence of attribute `PhysicalDimensionMapping.secondPhysicalDimension` ⌈For each `PhysicalDimensionMapping`, attribute `secondPhysicalDimension` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1917] Existence of `ConstantSpecification.valueSpec` ⌈For each `ConstantSpecification`, the aggregation of `ValueSpecification` in the role `valueSpec` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1918] Existence of `RecordValueSpecification.field` ⌈For each `RecordValueSpecification`, the aggregation of `ValueSpecification` in the role `field` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1919] **Existence of `TextValueSpecification.value`** ⌈For each `TextValueSpecification`, attribute `value` shall exist **at the time when the contract phase generation is executed.**

⌋*()*

[constr_1920] **Existence of `NumericalValueSpecification.value`** ⌈For each `NumericalValueSpecification`, attribute `value` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1921] **Existence of `ReferenceValueSpecification.referenceValue`** ⌈For each `ReferenceValueSpecification`, attribute `referenceValue` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1922] **Existence of `ApplicationRuleBasedValueSpecification.category`** ⌈For each `ApplicationRuleBasedValueSpecification`, attribute `category` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1923] **Existence of `RuleBasedAxisCont.ruleBasedValues`** ⌈For each `RuleBasedAxisCont`, attribute `ruleBasedValues` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1924] **Existence of `RuleBasedValueCont.ruleBasedValues`** ⌈For each `RuleBasedValueCont`, attribute `ruleBasedValues` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1925] **Existence of `NumericalRuleBasedValueSpecification.ruleBasedValues`** ⌈For each `NumericalRuleBasedValueSpecification`, attribute `ruleBasedValues` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1926] **Existence of `RuleBasedValueSpecification.rule`** ⌈For each `RuleBasedValueSpecification`, attribute `rule` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1927] **Existence of `RuleBasedValueSpecification.arguments`** ⌈For each `RuleBasedValueSpecification`, the aggregation of `RuleArguments` in the

role `arguments` shall exist **at the time when the contract phase genera-tion is executed**.

⌋*()*

[constr_1928] Existence of `CompositeRuleBasedValueSpecification.rule` ⌈For each `CompositeRuleBasedValueSpecification`, attribute `rule` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1929] Existence of `CompositeRuleBasedValueSpecification.argument` ⌈For each `CompositeRuleBasedValueSpecification`, the aggregation of `CompositeValueSpecification` in the role `argument` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1930] Existence of `ConstantReference.constant` ⌈For each `ConstantReference`, attribute `constant` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1931] Existence of `ConstantSpecificationMapping.applConstant` ⌈For each `ConstantSpecificationMapping`, the reference to meta-class `ConstantSpecification` in the role `applConstant` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1932] Existence of `ConstantSpecificationMapping.implConstant` ⌈For each `ConstantSpecificationMapping`, the reference to meta-class `ConstantSpecification` in the role `implConstant` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1933] Existence of `CalibrationParameterValue.initializedParameter` ⌈For each `CalibrationParameterValue`, the reference to meta-class `ConstantSpecification` in the role `initializedParameter` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1934] Existence of attribute `SwcInternalBehavior.handleTerminationAndRestart` ⌈For each `SwcInternalBehavior`, attribute `handleTerminationAndRestart` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1935] Existence of attribute `SwcInternalBehavior.supportsMultipleInstantiation` ⌈For each `SwcInternalBehavior`, attribute `supportsMul-

tipleInstantiation shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1936] **Existence of attribute RunnableEntity.symbol** ⌈For each RunnableEntity, attribute symbol shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1938] Existence of attribute RunnableEntityArgument.symbol ⌈For each RunnableEntityArgument, attribute symbol shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1939] **Existence of attribute ExecutableEntityActivationReason.bitPosition** ⌈For each ExecutableEntityActivationReason, attribute bitPosition shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1940] **Existence of attribute AsynchronousServerCallReturnsEvent.eventSource** ⌈For each AsynchronousServerCallReturnsEvent, attribute eventSource shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1941] Existence of attribute DataSendCompletedEvent.eventSource ⌈For each DataSendCompletedEvent, attribute eventSource shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1942] Existence of attribute DataWriteCompletedEvent.eventSource ⌈For each DataWriteCompletedEvent, attribute eventSource shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1943] **Existence of attribute DataReceivedEvent.data** ⌈For each DataReceivedEvent, attribute data shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1944] Existence of attribute `DataReceiveErrorEvent.data` ⌈For each `DataReceiveErrorEvent`, attribute `data` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1945] Existence of attribute `OperationInvokedEvent.operation` ⌈For each `OperationInvokedEvent`, attribute `operation` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1946] Existence of attribute `SwcModeSwitchEvent.activation` ⌈For each `SwcModeSwitchEvent`, attribute `activation` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1947] Existence of reference `SwcModeSwitchEvent.mode` ⌈For each `SwcModeSwitchEvent`, the reference to `ModeDeclaration` in the role `mode` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1948] Existence of attribute `ModeSwitchedAckEvent.eventSource` ⌈For each `ModeSwitchedAckEvent`, attribute `eventSource` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1949] Existence of attribute `ExternalTriggerOccurredEvent.trigger` ⌈For each `ExternalTriggerOccurredEvent`, attribute `trigger` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1950] Existence of attribute `InternalTriggerOccurredEvent.eventSource` ⌈For each `InternalTriggerOccurredEvent`, the attribute `eventSource` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1951] Existence of attribute `WaitPoint.timeout` ⌈For each `WaitPoint`, attribute `timeout` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1952] Existence of reference `WaitPoint.trigger` ⌈For each `WaitPoint`, the reference to `RTEEvent` in the role `trigger` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1953] Existence of attribute `SwcExclusiveAreaPolicy.apiPrinciple`
⌈For each `SwcExclusiveAreaPolicy` that refers to an `exclusiveArea`, attribute `apiPrinciple` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1954] Existence of attribute `VariableAccess.accessedVariable` ⌈For each `VariableAccess`, attribute `accessedVariable` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1955] Existence of attribute `ServerCallPoint.operation` ⌈For each `ServerCallPoint`, attribute `operation` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1956] Existence of attribute `ServerCallPoint.timeout` ⌈For each `ServerCallPoint`, attribute `timeout` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1957] Existence of attribute `AsynchronousServerCallResultPoint.asynchronousServerCallPoint` ⌈For each `AsynchronousServerCallResultPoint`, the reference to `AsynchronousServerCallPoint` in the role `asynchronousServerCallPoint` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1958] Existence of attribute `ParameterAccess.accessedParameter` ⌈For each `ParameterAccess`, attribute `accessedParameter` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1959] Existence of attribute `InstantiationDataDefProps.swDataDefProps` ⌈For each `InstantiationDataDefProps`, attribute `swDataDefProps` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1960] Existence of attribute `PortAPIOption.port` ⌈For each `PortAPIOption`, attribute `port` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1961] Existence of attribute `PortDefinedArgumentValue.value` ⌈For each `PortDefinedArgumentValue`, attribute `value` shall exist **at the time when the RTE is generated.**

⌋*()*

[constr_1962] Existence of attribute `PortDefinedArgumentValue.valueType` ⌈For each `PortDefinedArgumentValue`, attribute `valueType` shall exist **at the time when the RTE is generated.**

⌋*()*

[constr_1963] Existence of attribute `CommunicationBufferLocking.support-BufferLocking` ⌈For each `CommunicationBufferLocking`, attribute `support-BufferLocking` shall exist **at the time when the RTE is generated.**

⌋*()*

[constr_1964] Existence of attribute `PerInstanceMemory.type` ⌈For each `PerInstanceMemory`, attribute `type` shall exist **at the time when the contract phase generation is executed.**

⌋*()*

[constr_1965] Existence of attribute `PerInstanceMemory.typeDefinition` ⌈For each `PerInstanceMemory`, attribute `typeDefinition` shall exist **at the time when the contract phase generation is executed.**

⌋*()*

[constr_1966] Existence of attribute `Implementation.swVersion` ⌈For each `Implementation`, attribute `swVersion` shall exist **at the time when the RTE is generated.**

⌋*()*

[constr_1967] Existence of attribute `Implementation.vendorId` ⌈For each `Implementation`, attribute `vendorId` shall exist **at the time when the RTE is generated.**

⌋*()*

[constr_1968] Existence of attribute `Implementation.codeDescriptor` ⌈For each `Implementation`, at least one aggregation of `Code` in the role `codeDescriptor` shall exist **at the time when the RTE is generated.**

⌋*()*

[constr_1969] Existence of attribute `SwcImplementation.behavior` ⌈For each `SwcImplementation`, attribute `behavior` shall exist **at the time when the RTE is generated.**

⌋*()*

[constr_1970] Existence of attribute `PerInstanceMemorySize.alignment` ⌈For each `PerInstanceMemorySize`, attribute `alignment` shall exist **at the time when the RTE is generated.**

⌋*()*

[constr_1971] Existence of attribute `PerInstanceMemorySize.perInstance-Memory` ⌈For each `PerInstanceMemorySize`, the reference to `PerInstanceMemory` in the role `perInstanceMemory` shall exist **at the time when the RTE is generated.**

⌋*()*

[constr_1972] Existence of attribute `PerInstanceMemorySize.size` ⌈For each `PerInstanceMemorySize`, attribute `size` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1973] Existence of attribute `ModeDeclarationGroup.initialMode` ⌈For each `ModeDeclarationGroup`, the reference to `ModeDeclaration` in the role `initialMode` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1974] Existence of attribute `ModeDeclarationGroup.modeDeclaration` ⌈For each `ModeDeclarationGroup`, at least one `ModeDeclaration` shall be aggregated in the role `modeDeclaration` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_1975] Existence of attribute `ModeTransition.enteredMode` ⌈For each `ModeTransition`, the reference to `ModeDeclaration` in the role `enteredMode` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1976] Existence of attribute `ModeTransition.exitedMode` ⌈For each `ModeTransition`, the reference to `ModeDeclaration` in the role `exitedMode` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1977] Existence of attribute `ModeErrorBehavior.errorReactionPolicy` ⌈For each `ModeErrorBehavior`, the attribute `errorReactionPolicy` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1978] Existence of attribute `SwcModeManagerErrorEvent.modeGroup` ⌈For each `SwcModeManagerErrorEvent`, the instance reference to `ModeDeclara-`

`tion` in the role `modeGroup` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1979] Existence of the reference `SwcBswMapping.bswBehavior` ⌈For each `SwcBswMapping`, the reference to `BswInternalBehavior` in the role `bswBehavior` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1980] Existence of the reference `SwcBswMapping.swcBehavior` ⌈For each `SwcBswMapping`, the reference to `BswInternalBehavior` in the role `swcBehavior` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1981] Existence of attribute `NvBlockDescriptor.nvBlockNeeds` ⌈For each `NvBlockDescriptor`, attribute `nvBlockNeeds` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1982] Existence of attribute `ModeSwitchEventTriggeredActivity.role` ⌈For each `ModeSwitchEventTriggeredActivity`, attribute `role` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1983] Existence of attribute `ModeSwitchEventTriggeredActivity.swcModeSwitchEvent` ⌈For each `ModeSwitchEventTriggeredActivity`, attribute `swcModeSwitchEvent` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1984] Existence of instance reference `NvBlockDataMapping.nvRamBlockElement` ⌈For each `NvBlockDataMapping`, the instance reference to `ModeDeclaration` in the role `nvRamBlockElement` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1985] Existence of the reference `SupervisedEntityNeeds.toleratedFailedCycles` ⌈For each `SupervisedEntityNeeds`, the reference to `BswInternalBehavior` in the role `toleratedFailedCycles` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1986] Existence of the reference `DiagnosticRoutineNeeds.diagRoutineType` ⌈For each `DiagnosticRoutineNeeds`, the reference to `BswInternal-`

Behavior in the role diagRoutineType shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1987] Existence of instance reference RapidPrototypingScenario.hostSystem ⌈For each RapidPrototypingScenario, the instance reference to ModeDeclaration in the role hostSystem shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1988] Existence of attribute RptProfile.maxServicePointId ⌈For each RptProfile, attribute maxServicePointId shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1989] Existence of attribute RptProfile.minServicePointId ⌈For each RptProfile, attribute minServicePointId shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1990] Existence of attribute RptProfile.servicePointSymbolPost ⌈For each RptProfile, attribute servicePointSymbolPost shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1991] Existence of attribute RptProfile.servicePointSymbolPre ⌈For each RptProfile, attribute servicePointSymbolPre shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1992] Existence of attribute RptProfile.stimEnabler ⌈For each RptProfile, attribute stimEnabler shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1993] Existence of attribute RptImplPolicy.rptEnablerImplType ⌈For each RptImplPolicy, attribute rptEnablerImplType shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1994] Existence of attribute RptImplPolicy.rptPreparationLevel ⌈For each RptImplPolicy, attribute rptPreparationLevel shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1995] Existence of attribute `RptSwPrototypingAccess.rptHookAccess` ⌈For each `RptSwPrototypingAccess`, attribute `rptHookAccess` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1996] Existence of attribute `RptSwPrototypingAccess.rptReadAccess` ⌈For each `RptSwPrototypingAccess`, attribute `rptReadAccess` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1997] Existence of attribute `RptSwPrototypingAccess.rptWriteAccess` ⌈For each `RptSwPrototypingAccess`, attribute `rptWriteAccess` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1998] Existence of attribute `RptExecutableEntityProperties.maxRptEventId` ⌈For each `RptExecutableEntityProperties`, attribute `maxRptEventId` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_1999] Existence of attribute `RptExecutableEntityProperties.minRptEventId` ⌈For each `RptExecutableEntityProperties`, attribute `minRptEventId` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_2000] Compatibility of `ClientServerOperation`s triggering the same `RunnableEntity` ⌈The `ClientServerOperation`s are considered compatible **at the time when the contract phase generation is executed** if

- the number of `argument`s (which can be `ArgumentDataPrototype`s or related `PortDefinedArgumentValue`s) is equal and

- the corresponding `argument`s (i.e. first `argument` on both sides, second `argument` on both sides, etc.) are compatible or both are typed by "new-world" `Variable-Size Array Data Type`s where the data types of the array elements are compatible (but the array sizes may differ).

- and the respective values of `PortAPIOption.errorHandling` are identical.

In particular, this means that:

- for combinations of `ArgumentDataPrototype`s and `ArgumentDataPrototype`s where the `serverArgumentImplPolicy` is set to `useArgumentType` the referred `ImplementationDataType`s shall be compatible.

  In case of data types of `category STRUCTURE` all by order matching `ImplementationDataTypeElement`s shall be named equally.

- for combinations of `PortDefinedArgumentValue`s and `ArgumentDataPrototype`s where the `serverArgumentImplPolicy` is set to `useArgumentType` the referred `ImplementationDataType`s shall be compatible.

  In case of `ImplementationDataTypeElement`s of `category STRUCTURE` all by order matching `ImplementationDataTypeElement`s of the structure shall be named equally.

- for `ArgumentDataPrototype`s where the `serverArgumentImplPolicy` is set to `useVoid` an arbitrary `ImplementationDataType` is referred to.

In addition, it is required that the **return value defined on both sides shall match** (in terms of `Std_ReturnType` vs. `void`) and also the `possibleError`s are compatible.

⌋*()*

[constr_2002] **Referenced `VariableDataPrototype` from `AutosarVariableRef` of `VariableAccess` in role `dataReadAccess`** ⌈A `VariableAccess` in the role `dataReadAccess` shall refer to an `RPortPrototype` or `PRPortPrototype` that is typed by either a `SenderReceiverInterface` or a `NvDataInterface`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_2003] **Referenced `VariableDataPrototype` from `AutosarVariableRef` of `VariableAccess` in role `dataWriteAccess`** ⌈A `VariableAccess` in the role `dataWriteAccess` shall refer to a `PPortPrototype` or `PRPortPrototype` that is typed by either a `SenderReceiverInterface` or a `NvDataInterface`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_2004] **Referenced `VariableDataPrototype` from `AutosarVariableRef` of `VariableAccess` in role `dataSendPoint`** ⌈A `VariableAccess` in the role `dataSendPoint` shall refer to a `PPortPrototype` or `PRPortPrototype` that is typed by either a `SenderReceiverInterface` or a `NvDataInterface` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_2005] **Referenced `VariableDataPrototype` from `AutosarVariableRef` of `VariableAccess` in role `dataReceivePointByValue` or `dataReceivePointByArgument`** ⌈A `VariableAccess` in the role `dataReceivePointByValue` or `dataReceivePointByArgument` shall refer to an `RPortPrototype` or `PRPortPrototype` that is typed by either a `SenderReceiverInterface`

or an `NvDataInterface` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_2006] Number of `AsynchronousServerCallResultPoint` referencing to one `AsynchronousServerCallPoint` ⌈The `AsynchronousServerCallPoint` may be referenced by at most one `AsynchronousServerCallResultPoint` **at the time when the contract phase generation is executed**.

If the reference exists, this means that only the `RunnableEntity` with this `AsynchronousServerCallResultPoint` can fetch the result of the asynchronous server invocation of this particular `AsynchronousServerCallPoint`.

⌋*()*

[constr_2007] Consistency of `typeDefinition` attribute ⌈All `PerInstanceMemory`s of the same `SwcInternalBehavior` with identical `type` attribute shall define an identical `typeDefinition` attribute as well.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_2009] Supported kinds of `PortPrototype`s of a `NvBlockSwComponentType` ⌈With respect to external communication, `NvBlockSwComponentType` is limited to the definition of the following kinds of `PortPrototype`:

- `PortPrototype`s typed by either `NvDataInterface`s or `ClientServerInterface`s

- `RPortPrototype`s typed by `ModeSwitchInterface`s

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_2010] Connections between `SwComponentPrototype`s of type `NvBlockSwComponentType` ⌈The existence of `SwConnector`s that refer to `PortPrototype`s belonging to `SwComponentPrototype`s where both are typed by `NvBlockSwComponentType` is not permitted **at the time when the RTE is generated**.

⌋*()*

[constr_2011] Connections between `SwComponentPrototype`s typed by `NvBlockSwComponentType` and `SwComponentPrototype`s typed by other `AtomicSwComponentType`s ⌈A `PortPrototype` typed by an `NvDataInterface` owned by a `SwComponentPrototype` typed by an `NvBlockSwComponentType` shall be connected to a `PortPrototype` typed by **either** an `NvDataInterface` **or**

a `SenderReceiverInterface` owned by a `SwComponentPrototype` that is typed by an other subclass of `AtomicSwComponentType`.

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_2012] Compatibility of `ImplementationDataType`s used for `ramBlock` and `romBlock` ⌈The `ramBlock` and the `romBlock` shall have compatible `ImplementationDataType`s to ensure **at the time when the RTE is generated**, that the `NVRAM Block` default values in the `ROM Block` can be copied into the `RAM Block`.

⌋*()*

[constr_2013] Compatibility of `ImplementationDataType`s for `NvBlockDataMapping` ⌈Unless both the attribute `bitfieldTextTableMaskNvBlockDescriptor` and attribute `bitfieldTextTableMaskPortPrototype` is defined in the context of a given `NvBlockDataMapping`, the `NvBlockDataMapping` is only valid if the `ImplementationDataType` of the referenced `VariableDataPrototype` or `ImplementationDataTypeElement` in the role `nvRamBlockElement` is compatible to the `ImplementationDataType` used to type the `DataPrototype` aggregated by `NvBlockDataMapping` in the role `writtenNvData`, `writtenReadNvData`, or `readNvData`.

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_2014] Limitation of `NvBlockDescriptor.clientServerPort.role` ⌈The value of attribute `NvBlockDescriptor`.`clientServerPort`.`role` shall be set to a valid name of one of the Standardized AUTOSAR (client/server) Interfaces used for the NVRAM Manager, as described by [TPS_SWCT_02501], [TPS_SWCT_02502], [TPS_SWCT_02503] and [TPS_SWCT_02504].

⌋*()*

[constr_2015] Limitation of `SwcInternalBehavior` of a `NvBlockSwComponentType` ⌈The `SwcInternalBehavior` of a `NvBlockSwComponentType` is only permitted to define

- `OperationInvokedEvent`s

- `RunnableEntity`s triggered by `OperationInvokedEvent`s (server `RunnableEntity`s)

- `RunnableEntity`s which defines only the mandatory attributes `symbol` and `canBeInvokedConcurrently`

- `PortAPIOption`s defining `PortDefinedArgumentValue`s

- `TimingEvent`s (which may include references to `ModeDeclaration`s in the role `disabledMode`)

- `DataReceivedEvent`s (which may include references to `ModeDeclaration`s in the role `disabledMode`)

- `SwcModeSwitchEvent`s

- `RunnableEntity`s triggered by `TimingEvent`s

- `RunnableEntity`s triggered by `DataReceivedEvent`s

- `RunnableEntity`s triggered by `SwcModeSwitchEvent`s

- `DataTypeMappingSet`

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_2016] Connections between `SwComponentPrototype`s of type `ServiceProxySwComponentType` ⌈A connection between `PortPrototype`s belonging to `SwComponentPrototype`s where both are typed by `ServiceProxySwComponentType` is not permitted **at the time when the RTE is generated**.

⌋*()*

[constr_2017] Ports of `ServiceProxySwComponentType`s ⌈`ServiceProxySwComponentType` is only permitted to define

- `RPortPrototype`s that are typed by `SenderReceiverInterface` or

- `PortPrototype`s that are typed by a `PortInterface` where the `isService` attribute is set to true.

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_2018] Supported remote communication of a `ServiceProxySwComponentType` ⌈For remote communication, `ServiceProxySwComponentType` can have only `RPortPrototype`s typed by `SenderReceiverInterface`s in a 1:n communication scenario.

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_2019] `ServiceSwComponentType` shall have service ports only ⌈In the case of `ServiceSwComponentType`, all aggregated `PortPrototype`s need to have an ≪isOfType≫ relationship to a `PortInterface` which has its `isService` attribute set to `true` **at the time when the RTE is generated**.

The exceptions described in

- [TPS_SWCT_01572],

- [TPS_SWCT_01579],

- [TPS_SWCT_01831] and

- [TPS_SWCT_01580]

apply.

⌋*()*

[constr_2020] `dataReadAccess` can not be used for queued communication ⌈The `swImplPolicy` of the `VariableDataPrototype` referenced by a `VariableAccess` in role `dataReadAccess` shall **not** be set to `queued` **at the time when the contract phase generation is executed.**

⌋*()*

[constr_2021] `WaitPoint` referencing a `DataReceivedEvent` can not be used for non-queued communication ⌈A `WaitPoint` referencing a `DataReceivedEvent` is permitted **if and only if** the `swImplPolicy` of the `VariableDataPrototype` referenced by this `DataReceivedEvent` is set to `queued`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_2022] Mutually exclusive use of `SynchronousServerCallPoint`s and `AsynchronousServerCallPoint`s ⌈A `ClientServerOperation` of a particular `RPortPrototype` shall be mutually exclusive referenced by either a `SynchronousServerCallPoint`s or an `AsynchronousServerCallPoint`s **at the time when the contract phase generation is executed**.

⌋*()*

[constr_2023] Consistency of `timeout` values ⌈The `timeout` values of all `ServerCallPoint`s referencing the same instance of `ClientServerOperation` in a `RPortPrototype` shall be identical **at the time when the RTE is generated**.

⌋*()*

[constr_2024] `enableTakeAddress` is restricted to single instantiation ⌈The definition of a `PortAPIOption` with `enableTakeAddress` set to `true` is only permitted for software-components where the attribute `SwcInternalBehavior.supportsMultipleInstantiation` is set to `false` **at the time when the contract phase generation is executed.**

⌋*()*

[constr_2026] Referenced `VariableDataPrototype` from `AutosarVariableRef` of `VariableAccess` in role `writtenLocalVariable` and `readLocalVariable` ⌈A `VariableDataPrototype` in the `localVariable` reference needs to be owned by the same `SwcInternalBehavior` as this `RunnableEntity` belongs

to, and the referenced `VariableDataPrototype` has to be defined in the role `implicitInterRunnableVariable` or `explicitInterRunnableVariable`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_2027] `SwcServiceDependency` **shall be defined for service ports only** ⌈A `PortPrototype` that is referenced by a `SwcServiceDependency` via `assignedPort` or via `assignedData` shall be typed by a `PortInterface` that has `isService` set to `true` **at the time when the RTE is generated**.

This rule does **not** apply to `PortPrototype`s referenced by a `RoleBasedPortAssignment` where the attribute `role` is set to any of the following values:

- `NvMService`
- `NvMNotifyJobFinished`
- `NvMNotifyInitBlock`
- `NvMAdmin`
- `NvMMirror`
- `NvDataPort`

Furthermore, the rule does **not** apply to the case described in [TPS_SWCT_01579], [TPS_SWCT_01831], [TPS_SWCT_01580], and [TPS_SWCT_01572].

⌋*()*

[constr_2028] `staticMemory` **is restricted to single instantiation** ⌈The `staticMemory` is only supported if the attribute `supportsMultipleInstantiation` of the owning `SwcInternalBehavior` is set to `false` **at the time when the RTE is generated**

⌋*()*

[constr_2030] `AsynchronousServerCallResultPoint` **combined with** `WaitPoint` **shall belong to the same** `RunnableEntity` ⌈A `WaitPoint` referencing a `AsynchronousServerCallReturnsEvent` as well as a `AsynchronousServerCallResultPoint` referenced by said `AsynchronousServerCallReturnsEvent` shall be aggregated by the same `RunnableEntity` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_2031] Value of `TimingEvent.period` **shall be greater than 0** ⌈Attribute `TimingEvent.period` shall exist and its value shall be greater than 0 **at the time when the RTE is generated**.

⌋*()*

[constr_2033] Timeout of `DataSendCompletedEvent` ⌈The `timeout` value of a `WaitPoint` associated with a `DataSendCompletedEvent` shall have the same value as the corresponding value of `TransmissionAcknowledgementRequest.` `timeout` **at the time when the RTE is generated**.

⌋*()*

[constr_2034] `SwAddrMethod` referenced by `RunnableEntity`s, `BswCalledEntity`s, or `BswSchedulableEntity`s ⌈`RunnableEntity`s, `BswCalledEntity`s, and `BswSchedulableEntity`s shall not reference a `SwAddrMethod` which attribute `memoryAllocationKeywordPolicy` is set to `addrMethodShortNameAndAlignment` **at the time when the RTE is generated**.

⌋*()*

[constr_2035] `swImplPolicy` for `VariableDataPrototype` in `SenderReceiverInterface` ⌈The overriding value of attribute `swImplPolicy` of a `VariableDataPrototype` owned by a `SenderReceiverInterface` shall be either `standard`, `queued`, or `measurementPoint`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_2036] `swImplPolicy` for `VariableDataPrototype` in `NvDataInterface` ⌈The overriding value of attribute `swImplPolicy` of a `VariableDataPrototype` owned by a `NvDataInterface` shall be `standard`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_2037] `swImplPolicy` for `VariableDataPrototype` in the role `ramBlock` ⌈The overriding value of attribute `swImplPolicy` of a `VariableDataPrototype` aggregated in the role `NvBlockDescriptor.ramBlock` shall be `standard`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_2038] `swImplPolicy` for `VariableDataPrototype` in the role `implicitInterRunnableVariable` ⌈The overriding value of attribute `swImplPolicy` of a `VariableDataPrototype` aggregated in the role `SwcInternalBehavior.implicitInterRunnableVariable` shall be `standard`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_2039] `swImplPolicy` for `VariableDataPrototype` in the role `explicitInterRunnableVariable` ⌈The overriding value of attribute `swImplPolicy` of a `VariableDataPrototype` aggregated in the role `SwcInternalBehavior.explicitInterRunnableVariable` shall be `standard`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_2040] `swImplPolicy` for `VariableDataPrototype` in the role `arTypedPerInstanceMemory` ⌈The overriding value of attribute `swImplPolicy` of a `VariableDataPrototype` aggregated in the role `SwcInternalBehavior.arTypedPerInstanceMemory` shall be `standard` or `measurementPoint`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_2041] `swImplPolicy` for `VariableDataPrototype` in the role `staticMemory` ⌈The overriding value of attribute `swImplPolicy` of a `VariableDataPrototype` aggregated in the role `InternalBehavior.staticMemory` shall be `standard` or `measurementPoint`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_2042] `swImplPolicy` for `ParameterDataPrototype` in `ParameterInterface` ⌈The overriding value of attribute `swImplPolicy` of a `ParameterDataPrototype` owned by a `ParameterInterface` shall be either `standard`, `const`, or `fixed`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_2043] `swImplPolicy` for `ParameterDataPrototype` in the role `romBlock` ⌈The overriding value of attribute `swImplPolicy` a `ParameterDataPrototype` aggregated in the role `NvBlockDescriptor.romBlock` shall be `standard`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_2044] `swImplPolicy` for `ParameterDataPrototype` in the role `sharedParameter` ⌈The overriding value of attribute `swImplPolicy` of a `ParameterDataPrototype` aggregated in the role `SwcInternalBehavior.sharedParameter` shall be `standard` or `const`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_2045] `swImplPolicy` **for** `ParameterDataPrototype` **in the role** `perInstanceParameter` ⌈The overriding value of attribute `swImplPolicy` of a `ParameterDataPrototype` in the role `SwcInternalBehavior.perInstanceParameter` shall be `standard` or `const`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_2046] `swImplPolicy` **for** `ParameterDataPrototype` **in the role** `constantMemory` ⌈The overriding value of attribute `swImplPolicy` of a `ParameterDataPrototype` aggregated in the role `InternalBehavior.constantMemory` shall be `standard`, `const`, or `fixed`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_2047] `swImplPolicy` **for** `ArgumentDataPrototype` ⌈The overriding value of attribute `swImplPolicy` of an `ArgumentDataPrototype` shall be `standard`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_2049] Different `ModeDeclarationGroup`**s shall have different** `shortName`**s.** ⌈A software component is not allowed to type multiple `PortPrototype`s with `ModeSwitchInterface`s where the contained `ModeDeclarationGroupPrototype`s are referencing `ModeDeclarationGroup`s with identical `shortName`s but different `ModeDeclaration`s **at the time when the contract phase generation is executed**

⌋*()*

[constr_2050] Mandatory information of a `SwAxisCont` ⌈If the attribute `swAxisCont` is defined for an `ApplicationValueSpecification` the `SwAxisCont` shall define one `swAxisIndex` value and one `swArraysize` value per dimension, even in the case when the owning `ApplicationValueSpecification` defines only the content of a single dimensional object like a CURVE.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_2051] Mandatory information of a `SwValueCont` ⌈If the attribute `swValueCont` is defined for an `ApplicationValueSpecification` the `SwValueCont` shall always define the attribute `swArraysize` if the `ApplicationValueSpecification` is of `category` `CURVE`, `MAP`, `CUBOID`, `CUBE_4`, `CUBE_5`, `COM_AXIS`, `RES_AXIS`, or `VAL_BLK`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_2052] Values of `swArraysize` and the number of values provided by `swValuesPhys` shall be consistent. ⌈`swValuesPhys` shall define as many values as the attribute `swArraysize` (if this attribute exists) defines.

In other words, in the bound model the number of descendants (`v`, or `vf`, or `vt`, or `vtf`) shall be identical to the number of elements of the related `DataPrototype` typed by an `ApplicationPrimitiveDataType`.

If several `swArraysize` values are provided, the values have to be multiplied in order to get the total number of `swValuesPhys` values.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_2053] Consistency between `role` `IUMPRNumerator` and `ObdRatioServiceNeeds.connectionType` ⌈If a `SwcServiceDependency` with a `ObdRatioServiceNeeds` is defined and the attribute `connectionType` of the contained `ObdRatioServiceNeeds` is set to `ObdRatioConnectionKindEnum.apiUse`, a `RoleBasedPortAssignment` with the `role` value `IUMPRNumerator` shall be defined.

If the attribute `connectionType` of the contained `ObdRatioServiceNeeds` is set to `ObdRatioConnectionKindEnum.observer`, the `role` value `IUMPRNumerator` is not applicable.

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_2054] Valid targets of `rptSystem` ⌈The `System` referenced in the role `rptSystem` shall be of `category` `RPT_SYSTEM`.

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_2055] Valid targets of `byPassPoint` and `rptHook` reference, depending on the value of attribute `category` ⌈

Category	Meaning	Specific properties
`SW_COMPONENT_ PROTOTYPE`	Adds one `SwComponentPrototype` to an Rapid Prototyping Scenario.	The `byPassPoint` and `rptArHook` shall reference a `SwComponentPrototype`s.
`DATA_PROTOTYPE`	Adds one instance of a `DataPrototype` to an Rapid Prototyping Scenario.	The `byPassPoint` and `rptArHook` shall reference a `DataPrototype` instances in `PortPrototype`s.
`RUNNABLE_ENTITY`	Adds one `RunnableEntity` to an Rapid Prototyping Scenario.	The `byPassPoint` and `rptArHook` shall reference a `RunnableEntity` instances.
`ACCESS_POINTS`	Adds one `VariableAccess`, `ParameterAccess`, `ServerCallPoint`, `AsynchronousServerCallResultPoint`, `InternalTriggeringPoint`, `ModeSwitchPoint`, `ModeAccessPoint` or `ExternalTriggeringPoint` to a Rapid Prototyping Scenario.	The `byPassPoint` and `rptArHook` shall reference a `VariableAccess`, `ParameterAccess`, `ServerCallPoint`, `AsynchronousServerCallResultPoint`, `InternalTriggeringPoint`, `ModeSwitchPoint`, `ModeAccessPoint` or `ExternalTriggeringPoint` instances.

⌋*()*

[constr_2056] Consistency of `RapidPrototypingScenario` with respect to `rptSystem` and `rptArHook` references ⌈Within one `RapidPrototypingScenario` all `rptSystem` references shall point to instances in one and only one `System`, and if existent, all `rptArHook` shall point to instances in one other and only one other `System`.

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_2057] Mandatory information of a `RuleBasedAxisCont` ⌈If the attribute `swAxisCont` is defined for an `ApplicationRuleBasedValueSpecification` the `RuleBasedAxisCont` shall define one `swAxisIndex` value and one `swArraysize` value per dimension, even in the case when the owning `ApplicationRuleBasedValueSpecification` defines only the content of a single dimensional object like a `CURVE`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_2058] Mandatory information of a `RuleBasedValueCont` ⌈If the attribute `swValueCont` is defined for an `ApplicationRuleBasedValueSpecification` the `RuleBasedValueCont` shall always define the attribute `swArraysize` if the `ApplicationRuleBasedValueSpecification` is of `category CURVE`, `MAP`, `CUBOID`, `CUBE_4`, `CUBE_5`, `COM_AXIS`, `RES_AXIS`, or `VAL_BLK`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_2535] Target of an `autosarParameter` in `AutosarParameterRef` shall refer to a parameter ⌈Except for the specifically described cases where [constr_1173], applies the target of `autosarParameter` (which in fact is an instance ref) in `AutosarParameterRef` shall either be or be nested in `ParameterDataPrototype`. This means that the target shall either be a `ParameterDataPrototype` or an `ApplicationCompositeElementDataPrototype` that in turn is owned by a `ParameterDataPrototype`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_2536] Target of an `autosarVariable` in `AutosarVariableRef` shall refer to a variable ⌈The target of `autosarVariable` (which in fact is an instance ref) in `AutosarVariableRef` shall either be or be nested in `VariableDataPrototype`. This means that the target shall either be a `VariableDataPrototype` or an `ApplicationCompositeElementDataPrototype` that in turn is owned by a `VariableDataPrototype`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_2544] Limits need to be consistent ⌈

- The limits of `ApplicationDataType` shall be inside the definition range of the `CompuMethod`

  The `CompuMethod` needs to be applicable for limits of an `ApplicationDataType`. The reason is that the internal representation of the limits for the `ApplicationDataType` are calculated by applying the `CompuMethod`.

- The such defined internal limits of the `ApplicationDataType` shall be within or equal the `internalConstrs` of the mapped `ImplementationDataType`.

- The limits of the `ImplementationDataType` shall be within or equal to the limits defined by the size of the `BaseType`.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_2545] `invalidValue` shall fit in the specified ranges ⌈The `invalidValue` shall be in the range of the `ImplementationDataType` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_2548] Data constraint of value axis shall match ⌈The values compliant to `SwDataDefProps.dataConstr` shall also be compliant to `SwDataDefProps.valueAxisDataType.swDataDefProps.dataConstr`.

In other words `SwDataDefProps.dataConstr` win over but are not allowed to relax `SwDataDefProps.valueAxisDataType.swDataDefProps.dataConstr` but are not allowed.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_2549] Units of input axis shall be consistent ⌈The units specified in the context of an input axis shall be compatible, even if there is a precedence rule.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_2550] Units of value axis shall be consistent ⌈The units specified in the context of value axis shall be the same, even if there is a precedence rule.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_2561] Application of `DataConstrRule.constrLevel` ⌈`DataConstrRule.constrLevel` is limited to

0: This represents so called "hard limits". They shall always be specified.

1: This represents so called "soft limits". Soft limits may be violated after confirmation by the user of an MCD-System.

This rule applies **at any time in the workflow**. Other values may exist, but the semantics is outside the AUTOSAR scope.

⌋*()*

[constr_3688] Existence of attribute `RoleBasedDataAssignment.usedDataElement.localVariable` for `RoleBasedDataAssignment.role` = Server ServiceOffer ⌈If the attribute `RoleBasedDataAssignment.role` is set to the value `ServerServiceOffer`, then the reference `RoleBasedDataAssignment.usedDataElement.localVariable` shall not exist **at the time when the RTE is generated**.

⌋*()*

[constr_3689] Existence of attribute `RoleBasedDataAssignment.usedDataElement.localVariable` for `RoleBasedDataAssignment.role` = Client EventSubscription ⌈If the attribute `RoleBasedDataAssignment.role` is set

to the value `ClientEventSubscription`, then the reference `RoleBased-DataAssignment.usedDataElement.localVariable` shall not exist **at the time when the RTE is generated**.

⌋*()*

[constr_4002] Unambiguous mapping of modes to data types ⌈Within one `DataTypeMappingSet`, a `ModeDeclarationGroup` shall not be mapped to different `ImplementationDataType`s **at the time when the contract phase generation is executed**.

⌋*()*

[constr_4003] Semantics of `SwcModeSwitchEvent` ⌈If the value of `SwcModeSwitchEvent.activation` is `onTransition`, then `SwcModeSwitchEvent` shall refer to two different `ModeDeclaration`s belonging to the same instance of `ModeDeclarationGroup`.

Their order defines the direction of the transition from one mode into another. In all other cases `SwcModeSwitchEvent` shall refer to exactly one `ModeDeclaration`.

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_4004] Context of `SenderReceiverAnnotation` ⌈A `SenderReceiverAnnotation` shall only be aggregated by a `PortPrototype` typed by a `SenderReceiverInterface` **at any time in the workflow**.

⌋*()*

[constr_4005] Context of `ClientServerAnnotation` ⌈A `ClientServerAnnotation` shall only be aggregated by a `PortPrototype` typed by a `ClientServerInterface` **at any time in the workflow**.

⌋*()*

[constr_4006] Context of `ParameterPortAnnotation` ⌈A `ParameterPortAnnotation` shall only be aggregated by a `PPortPrototype` owned by a `ParameterSwComponentType` **at any time in the workflow**.

⌋*()*

[constr_4007] Context of `ModePortAnnotation` ⌈A `ModePortAnnotation` shall only be aggregated by a `PortPrototype` typed by a `ModeSwitchInterface` **at any time in the workflow**.

⌋*()*

[constr_4008] Context of `TriggerPortAnnotation` ⌈A `TriggerPortAnnotation` shall only be aggregated by a `PortPrototype` typed by a `TriggerInterface` **at any time in the workflow**.

⌋*()*

[constr_4009] Context of `NvDataPortAnnotation` ⌈An `NvDataPortAnnotation` shall only be aggregated by a `PortPrototype` typed by an `NvDataInterface` **at any time in the workflow**.

⌋*()*

[constr_4010] Context of `DelegatedPortAnnotation` ⌈A `DelegatedPortAnnotation` shall only be aggregated by a `PortPrototype` aggregated by a `CompositionSwComponentType`.

⌋*()*

[constr_4012] Timeout of `ModeSwitchedAckEvent` ⌈The timeout value of a `WaitPoint` associated with a `ModeSwitchedAckEvent` shall be equal to the corresponding `ModeSwitchedAckRequest.timeout` **at the time when the RTE is generated**.

⌋*()*

[constr_4082] `RunnableEntity.reentrancyLevel` shall not be set. ⌈The optional attribute `reentrancyLevel` shall not be set for a `RunnableEntity`. This attribute would define more specific reentrancy features than the mandatory attribute `canBeInvokedConcurrently`. These features are currently only supported for Basic Software.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_5234] Existence of attribute `E2EProfileCompatibilityProps.transitToInvalidExtended` is mandatory for each `EndToEndTransformationComSpecProps` ⌈For each `EndToEndTransformationComSpecProps`, a reference in the role `e2eProfileCompatibilityProps` to meta-class `E2EProfileCompatibilityProps` shall exist and the referenced `E2EProfileCompatibilityProps` shall define a value for the attribute `transitToInvalidExtended` **at the time when the RTE is generated**.

⌋*()*

[constr_10000] Existence of attribute `RptExecutableEntityProperties.rptExecutionControl` ⌈For each `RptExecutableEntityProperties`, attribute `rptExecutionControl` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_10001] Existence of attribute `RptExecutableEntityProperties.rptServicePoint` ⌈For each `RptExecutableEntityProperties`, attribute `rptServicePoint` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_10005] Existence of attribute `NotAvailableValueSpecification.defaultPattern` ⌈For each `NotAvailableValueSpecification`, attribute `defaultPattern` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_10006] Valid interval of attribute `NotAvailableValueSpecification.defaultPattern` ⌈The valid interval for attribute `NotAvailableValueSpecification.defaultPattern` **at the time when the contract phase generation is executed** is 0..255.

⌋*()*

[constr_10009] Aggregation of `ApplicationRuleBasedValueSpecification` ⌈Each `ArrayValueSpecification` shall only aggregate at most one `ApplicationRuleBasedValueSpecification` in the role element.

If one `ApplicationRuleBasedValueSpecification` is aggregated then it shall be the only aggregated element, i.e. no further `ValueSpecification` shall exist in the same aggregation where an `ApplicationRuleBasedValueSpecification` is aggregated.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_10016] Applicability of `OsTaskExecutionEvent` ⌈An `OsTaskExecutionEvent` is only applicable for a `SwcInternalBehavior` in the context of a `ComplexDeviceDriverSwComponentType`, `EcuAbstractionSwComponentType`, or `ServiceSwComponentType` **at any time in the workflow**.

⌋*()*

[constr_10017] Existence of attribute `SwAxisCont.category` ⌈For each `SwAxisCont`, attribute `category` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_10018] Existence of attribute `SwAxisCont.swAxisIndex` ⌈For each `SwAxisCont`, attribute `swAxisIndex` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_10019] Existence of attribute `SwAxisCont.swValuesPhys` ⌈For each `SwAxisCont`, attribute `swValuesPhys` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_10020] Existence of attribute `RoleBasedDataTypeAssignment.usedImplementationDataType` ⌈For each `RoleBasedDataTypeAssignment`, attribute `usedImplementationDataType` shall exist **at the time when the RTE is generated**.

⌋*()*

[constr_10032] Restrictions for the usage of `ServiceDependency.diagnosticRelevance` ⌈The attribute `ServiceDependency.diagnosticRelevance` shall only be used for a `SwcServiceDependency` that aggregates a `BswMgrNeeds` **at the time when the RTE is generated**.

⌋*()*

[constr_10033] Existence of `MemorySection.swAddrmethod` ⌈For each `MemorySection`, attribute `swAddrmethod` shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_10034] Existence of `MemorySection.alignment` ⌈For each `MemorySection`, attribute `alignment` shall exist **at the time when the contract phase generation is executed** if the attribute `MemorySection.swAddrmethod.memoryAllocationKeywordPolicy` is set to `MemoryAllocationKeywordPolicyType.addrMethodShortNameAndAlignment`.

⌋*()*

[constr_10040] Value of `ApplicationValueSpecification.swAxisCont.category` ⌈The value of attribute `ApplicationValueSpecification.swAxisCont.category` shall not be set to `fixAXIS` **at the time when the contract phase generation is executed**.

⌋*()*

[constr_10041] Value of `ApplicationRuleBasedValueSpecification.swAxisCont.category` ⌈The value of `ApplicationValueSpecification.swAxisCont.category` shall not be set to `fixAXIS`

⌋*()*

[constr_10067] Creation of `AssemblySwConnector` for service communication ⌈If an `AssemblySwConnector` is created between two PortPrototypes and the affected `PortInterface`s set the attribute `isService` to the value `true`, then at least one of the `SwComponentPrototype`s shall be typed by a `ServiceSwComponentType`. This constraint shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_10068] Standardized values for `SectionInitializationPolicyType` ⌈The following values for `SectionInitializationPolicyType` are reserved by the AUTOSAR standard:

INIT To be used for (explicitly or not explicitly) initialized variables.

CLEARED To be used for (explicitly or not explicitly) initialized variables.

POWER-ON-CLEARED To be used for variables that are not explicitly initialized (cleared) during normal start-up. Instead these are cleared only after power on reset.

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_10071] Allowed multiplicities of `SenderComSpec` attributes for communication between `ApplicationSwComponentType` and `NvBlockSwComponentType` ⌈

Sender	ApplicationSwComponentType	
Receiver	NvBlockSwComponentType	
Queuing Configuration	non-queued	queued
SenderComSpec.transmissionAcknowledge	d/c	n/a
SenderComSpec.dataElement	1	n/a
SenderComSpec.handleOutOfRange	d/c	n/a
SenderComSpec.usesEndToEndProtection	d/c	n/a
SenderComSpec.transmissionProps.dataUpdatePeriod	0..1	n/a
SenderComSpec.transmissionProps.minimumSendInterval	0..1	n/a
SenderComSpec.transmissionProps.transmissionMode	0..1	n/a
SenderComSpec.networkRepresentation	d/c	n/a
SenderComSpec.compositeNetworkRepresentation	d/c	n/a
NonqueuedSenderComSpec.dataFilter	d/c	n/a
NonqueuedSenderComSpec.initValue	0..1	n/a

⌋*()*

[constr_10072] Allowed multiplicities of `SenderComSpec` attributes for communication between `NvBlockSwComponentType` and `ApplicationSwComponentType` ⌈

Sender	NvBlockSwComponentType	
Receiver	ApplicationSwComponentType	
Queuing Configuration	non-queued	queued
ReceiverComSpec.replaceWith	0	n/a
ReceiverComSpec.dataElement	1	n/a
ReceiverComSpec.receptionProps.dataUpdatePeriod	0	n/a
ReceiverComSpec.receptionProps.timeout	0	n/a
ReceiverComSpec.usesEndToEndProtection	0	n/a
ReceiverComSpec.maxDeltaCounterInit	0	n/a

▽

△

Sender	**NvBlockSwComponentType**	
Receiver	**ApplicationSwComponentType**	
Queuing Configuration	non-queued	queued
ReceiverComSpec.handleOutOfRange	0	n/a
ReceiverComSpec.handleOutOfRangeStatus	0	n/a
ReceiverComSpec.maxNoNewOrRepeatedData	0	n/a
ReceiverComSpec.syncCounterInit	0	n/a
ReceiverComSpec.transformationComSpecProps	0	n/a
ReceiverComSpec.networkRepresentation	0	n/a
ReceiverComSpec.compositeNetworkRepresentation	0	n/a
QueuedReceiverComSpec.queueLength	n/a	n/a
NonqueuedReceiverComSpec.filter	0	n/a
NonqueuedReceiverComSpec.timeoutSubstitutionValue	0	n/a
NonqueuedReceiverComSpec.initValue	0..1	n/a
NonqueuedReceiverComSpec.aliveTimeout	0	n/a
NonqueuedReceiverComSpec.enableUpdate	0	n/a
NonqueuedReceiverComSpec.handleDataStatus	0	n/a
NonqueuedReceiverComSpec.handleNeverReceived	0..1	n/a
NonqueuedReceiverComSpec.handleTimeoutType	0	n/a

⌋*()*

[constr_10073] Existence of `DataReceiveErrorEvent` ⌈A `DataReceiveErrorEvent` shall only exist if it latest **at the time when the contract phase generation is executed** refers to a given `VariableDataPrototype` in the role `data` where either

- the `VariableDataPrototype` is referenced from a `NonqueuedReceiverComSpec` in the role `dataElement` and the attribute `aliveTimeout` of the `NonqueuedReceiverComSpec` exists and is set to a value > 0 **or**

- the `VariableDataPrototype` is aggregated by a `SenderReceiverInterface` where attribute `invalidationPolicy`.`handleInvalid` exists and is set to the value `keep`.

⌋*()*

[constr_10074] Consistency of attribute `NvBlockDescriptor`.`writingStrategy`.`role` set to `storeOnChange` ⌈The existence of `NvBlockDescriptor`.`writingStrategy` where attribute `role` is set to `storeOnChange` is only supported if `NvBlockDescriptor`.`nvBlockNeeds`.`storeOnChange` exists and is set to `true` **at the time when the RTE is generated**.

⌋*()*

[constr_10075] Existence of `CompositeRuleBasedValueSpecification`.`argument` vs. `compoundPrimitiveArgument` ⌈For every `CompositeRuleBasedValueSpecification`, at most one of the aggregations

- `argument`

• compoundPrimitiveArgument

shall exist **at the time when the contract phase generation is executed**.

⌋*()*

[constr_10087] Restriction for the existence of a SubElementMapping ⌈The existence of a DataPrototypeMapping.subElementMapping is only supported if the PortPrototypes that are referenced by the respective SwConnector are typed by a DataInterface **at the time when the RTE is generated**.

⌋*()*

[constr_10096] Shared axis shall not be a fixed axis ⌈An ApplicationPrimitiveDataType of category COM_AXIS shall not contain the definition of an SwCalprmAxis of category FIX_AXIS.

This rule shall be imposed **at the time when the contract phase generation is executed**

⌋*()*

[constr_10097] Buffer locking is only supported if returnValueProvision is set to returnValueProvided ⌈Setting the value of attribute PortAPIOption.supportedFeature.supportBufferLocking to value supportsBufferLocking is only supported if the AbstractAccessPoints that refer to values in the respective PortAPIOption.port do **not** define AbstractAccessPoint.returnValueProvision **or** set the value of AbstractAccessPoint.returnValueProvision to returnValueProvided.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_10099] Allowed values of the attribute `SwDataDefProps.swImplPolicy` vs. `DataPrototype`s and their roles ⌈

Attribute of SwImplPolicyEnum	VariableDataPrototype							ParameterDataPrototype					Misc.
	VariableDataPrototype in SenderReceiverInterface	VariableDataPrototype in NvDataInterface	VariableDataPrototype in role ramBlock	VariableDataPrototype in role implicitInterRunnableVariable	VariableDataPrototype in role explicitInterRunnableVariable	VariableDataPrototype in role arTypedPerInstanceMemory	VariableDataPrototype in role staticMemory	ParameterDataPrototype in ParameterInterface	ParameterDataPrototype in role romBlock	ParameterDataPrototype in role sharedParameter	ParameterDataPrototype in role perInstanceParameter	ParameterDataPrototype in role constantMemory	ArgumentDataPrototype
const	NA	NA	NA	NA	NA	NA	NA	x	NA	x	x	x	NA
fixed	NA	NA	NA	NA	NA	NA	NA	x	NA	NA	NA	x	NA
measurementPoint	x	NA	NA	NA	NA	x	x	NA	NA	NA	NA	NA	NA
queued	x	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
standard	x	x	x	x	x	x	x	x	x	x	x	x	x

⌋*()*

[constr_10104] `RoleBasedPortAssignment` where attribute `role` is set to `CallbackGetFaultDetectCounter` shall refer to a `PPortPrototype` in the role `portPrototype` ⌈If a `SwcServiceDependency` aggregates both

- a `DiagnosticEventNeeds` that in turn aggregates `DiagEventDebounceMonitorInternal` in the role `diagEventDebounceAlgorithm` and
- a `RoleBasedPortAssignment` in the role `assignedPort` where attribute `role` is set to `CallbackGetFaultDetectCounter`,

then the target of the reference `SwcServiceDependency.assignedPort.portPrototype` shall be a `PPortPrototype`.

This rule shall be imposed **at the time when the RTE is generated**.

⌋*()*

[constr_10118] Structural consistency of the modeling of `InvalidationPolicy` ⌈A `dataElement` referenced by an `InvalidationPolicy` shall be owned by the `SenderReceiverInterface` that also owns the `InvalidationPolicy` **at any time in the workflow**.

⌋*()*

[constr_10119] `SenderReceiverInterface`.dataElement shall be referenced by at most one `InvalidationPolicy` ⌈Any `SenderReceiverInterface`.

dataElement shall be referenced by at most one InvalidationPolicy in the role InvalidationPolicy.dataElement **at any time in the workflow**.

⌋*()*

[constr_10120] Structural consistency of the modeling of `MetaDataItemSet` ⌈A dataElement referenced by an MetaDataItemSet in the role dataElement shall be owned by the SenderReceiverInterface that also owns the MetaDataItem-Set **at any time in the workflow**.

⌋*()*

[constr_10121] `SenderReceiverInterface.dataElement` shall be referenced by at most one `MetaDataItemSet` ⌈Any SenderReceiverInterface.dataElement shall be referenced by at most one MetaDataItemSet in the role MetaDataItemSet.dataElement **at any time in the workflow**.

⌋*()*

[constr_10123] Existence of attribute `DtcStatusChangeNotificationNeeds.notificationTime` ⌈Attribute DtcStatusChangeNotificationNeeds.notificationTime shall only exist if the enclosing SwcServiceDependency contains a RoleBasedPortAssignment where attribute role is set to the value ClearDtcNotification **at the time when the RTE is generated**.

⌋*()*

[constr_10196] Definition of `invalidValue` for `DataPrototype` is typed by an `ImplementationDataType` that references a `CompuMethod` of `category` TEXTTABLE or BITFIELD_TEXTTABLE ⌈If an invalidValue is defined for a DataPrototype that is typed by an ImplementationDataType that references or inherits (see [constr_1015]) a CompuMethod of category TEXTTABLE or BITFIELD_TEXTTABLE, the applicable ValueSpecification shall be a TextValueSpecification **if** the value fits into the intervals defined by the CompuMethod.

In this case the value provided shall match to one of the applicable text values (vt, shortLabel, symbol) defined by the applicable CompuScales.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

[constr_10372] Relation between *Type of `PortPrototype`*, *Type of `ComSpec`*, and *Type of `PortInterface`* ⌈With respect to [constr_1043], if a *Type of PortPrototype* aggregates a *Type of ComSpec*, then the *Type of PortPrototype* shall

- reference a *Type of PortInterface* in the role *Role of Type-Ref* and

- the *Role of Element* that is referenced from the *Type of ComSpec* shall be aggregated by the exact same *Type of PortInterface* that is also referenced by the enclosing *Type of PortPrototype* in the role *Role of Type-Ref*.

⌋*()*

[constr_10373] `ImplementationDataType` of `category` `VALUE` shall not refer to `SwBaseType` of `category` `VOID` ⌈An `ImplementationDataType` where attribute `category` is set to `VALUE` shall not reference (in the role `swDataDefProps`.`base-Type`) a `SwBaseType` where attribute `category` is set to `VOID`.

⌋*()*

[constr_10383] Supported value encodings for `SwBaseType` in the context of `PortInterface`s where attribute `isService` is set to `false` ⌈The supported value encodings for the usage within a `PortInterface` where attribute `isService` is set to `false` are:

- `2C`: Two's complement

- `IEEE754`: floating-point numbers

- `ISO-8859-1`: single-byte coded character

- `ISO-8859-2`: single-byte coded character

- `WINDOWS-1252`: single-byte coded character

- `UTF-8`: UCS Transformation Format 8

- `UTF-16`: Character encoding for Unicode code points based on 16 bit code units [8]

- `UCS-2`: Universal Character Set 2

- `NONE`: Unsigned Integer

- `BOOLEAN`: This represents an integer to be interpreted as boolean.

This rule shall be imposed **at the time when the contract phase generation is executed**.

⌋*()*

2.7 TPS_StandardizationTemplate

[constr_2500] `PortInterface`s shall be of same kind ⌈Both objects (`PortInterface`s) referenced by a blueprint mapping for port interfaces (represented by `BlueprintMapping`) shall be of the same kind (e.g. both shall be `SenderReceiverInterface`s). In other words both interfaces shall be instances of the same meta class.

⌋*()*

[constr_2526] `PortInterface` **need to be compatible to the blueprints** ⌈`PortInterface` shall be compatible to their respective blueprints according to the compatibility rules.

⌋*()*

[constr_2527] Blueprints shall live in package of a proper category ⌈As explained in detail in the [9], model artifacts (in this case `PortPrototypeBlueprint` and incompletely specified `PortInterface`s) created for the purpose of becoming blueprints shall reside in an `ARPackage` of category `BLUEPRINT`.

⌋*()*

[constr_2528] `PortPrototype`**s shall not refer to blueprints of a** `PortInterface` ⌈A port `PortPrototype` shall not reference a `PortInterface` which lives in a package of category BLUEPRINT.

⌋*()*

[constr_2529] `PortPrototypeBlueprint`**s and derived** `PortPrototype`**s shall reference proper** `PortInterface`**s** ⌈A `PortPrototypeBlueprint` may reference a blueprint of `PortInterface`. According to [constr_2570], a system description shall not contain blueprints. Therefore the reference to the `PortInterface` may need to be rewritten when a `PortPrototype` is derived from the blueprint.

In this case the `PortInterface` referenced by the derived `PortPrototype` shall be compatible to the `PortInterface` (which is a blueprint) referenced by the `PortPrototypeBlueprint`.

According to [constr_2526] this can be ensured if the `PortInterface` referenced by the `PortPrototypeBlueprint` is the blueprint of the `PortInterface` referenced by the respective `PortPrototype`.

⌋*()*

[constr_2540] Tagged text category ⌈The `category` of `TraceableText` shall be one of

`ADVISORY_ITEM` The text represents a particular advisory. Such an item is applicable primarily in template specifications. It is similar to a constraint item but represents the characteristic of a WARNING rather than an ERROR.

`CONSTRAINT_ITEM` The text represents a particular constraint. Such an item is applicable primarily in template specifications. It is similar to a specification item but represents issues that may be validated automatically e.g. by a tool.

`IMPLEMENTATION_ITEM` The text represents a short description of an implementation. It is applicable primarily within the `introduction` of a model element.

`REQUIREMENT_ITEM` The text represents a particular requirement. Such an item is applicable primarily in requirement specifications.

SAFETY_* The text represents the type of safety requirements. The allowed values (*) are defined in `[TPS_SAFEX_00102]` in [**?**,].

SPECIFICATION_ITEM The text represents a particular item in the specification. Such an item is a requirement for the implementation of the software specification.

SRC The text represents the source code content.

TEST_ITEM The text represents a short description of a test. Such an item is applicable primarily in test specifications.

⌋*()*

[constr_2546] **References in derived model elements** ⌈Model elements derived from blueprints shall never refer to model elements that are blueprints.

⌋*()*

[constr_2553] `shortName` **shall follow the pattern defined in the Blueprint** ⌈The `shortName` respectively `symbol` of the derived objects shall follow the pattern defined in `namePattern` or `blueprintValue` of the blueprint according to [TPS_STDT_-00086]

⌋*()*

[constr_2554] **Derived objects shall match the blueprints** ⌈Unless specified explicitly otherwise, the attributes of the blueprint shall appear in the derived objects. As an exception `namePattern` and `blueprintValue` may **not** be copied.

⌋*()*

[constr_2556] **No Blueprint Motivated** `VariationPoint`**s in AUTOSAR Descriptions** ⌈AUTOSAR descriptions which are not blueprints shall not have `blueprint-Condition`, `formalBlueprintGenerator` nor `blueprintValue`.

⌋*()*

[constr_2563] `BswModuleDescription` **blueprints should not have a** `BswInternalBehavior` ⌈A `BswModuleDescription` blueprint should not have a `BswInternalBehavior` since this is a matter of implementation and not subject to standardization. Exceptions might exist in vendor internal applications.

⌋*()*

[constr_2564] `VariationPoint` **in Blueprints of** `PackageableElement` ⌈To support standardization, constraint [constr_2537] in [9] is relaxed for blueprints. This means in particular, that all `PackageableElement`s which inherit from `Atp-Blueprint` and live in a package of category BLUEPRINT may have a `VariationPoint`. In this case `vh.latestBindingTime` is considered as blueprint

`DerivationTime` even if the meta model still states `systemDesignTime` for `PackageableElement`.

⌋*()*

[constr_2565] Trace shall not be nested ⌈Due to the intended atomicity of requirements respectively specification items, `Traceable` shall not be nested.

⌋*()*

[constr_2566]   Blueprintmapping shall map appropriate elements ⌈ `BlueprintMapping` shall map elements which represent a valid pair of blueprint / derived object. In most of the cases this means that `blueprint` and `derivedObject` shall refer to objects of the same meta-class.

⌋*()*

[constr_2568] `SwComponentType`s shall be of same kind ⌈Both objects (`SwComponentType`s) referenced by a blueprint mapping for port interfaces (represented by `BlueprintMapping`) shall be of the same kind (e.g. both shall be `AtomicSwComponentType`s). In other words both components shall be instances of the same meta class.

⌋*()*

[constr_2569]   Purely Blueprint Motivated `VariationPoint`s ⌈`VariationPoint`s with `vh.latestBindingTime` set to `blueprintDerivationTime` shall have only `blueprintCondition` or `formalBlueprintGenerator` respectively `blueprintValue`.

⌋*()*

[constr_2570] No Blueprints in system descriptions ⌈There shall be no blueprints in system descriptions. In consequence of this blueprint elements shall be referenced only from blueprints and `AtpBlueprintMapping`s. Due to ≪atpUriDef≫, the references from `AtpBlueprintMapping` do not need to be resolved in system descriptions.

⌋*()*

[constr_2571] Outgoing references from Blueprints ⌈Note that outgoing references from Blueprints are basically not limited. Practically, references to objects living in a package of category EXAMPLE should not occur.

⌋*()*

[constr_2589] In VFB Timing Blueprint `TDEventVfbPort` shall reference `PortPrototypeBlueprint` ⌈In a VFB Timing Blueprint `TDEventVfbPort` shall reference `PortPrototypeBlueprint`. In other words, a VFB Timing Description Event specified in a VFB Timing Blueprint shall always reference a Port Prototype Blueprint.

⌋*()*

[constr_2590] One BlueprintPolicy is allowed ⌈For each attribute of a blueprint, at most one `BlueprintPolicy` is allowed.

⌋*()*

[constr_2591] BlueprintPolicyNotModifiable ⌈If `BlueprintPolicyNotModifiable` is assigned to an attribute, then during blueprinting it is not allowed to modify the value of the attribute and all its contained content.

⌋*()*

[constr_2592] No BlueprintPolicy ⌈If no `BlueprintPolicy` is assigned to an attribute, then arbitrary modifications are allowed while deriving from the blueprint.

⌋*()*

[constr_2593] Expression for identifying the attribute a BlueprintPolicy relates to ⌈The expression language for identifying the related attribute of a `BlueprintPolicy` is a subset version of xpath, see [10]. For navigation over the model we use the names as they are used in XML.

⌋*()*

[constr_2597] `ClientServerOperationBlueprintMapping` constrains number of arguments ⌈The number of arguments of the `BswModuleEntry` referenced by a `bswModuleEntry` shall be identical to the number of `portDefinedArgumentBlueprint`s of the owning `ClientServerInterfaceToBswModuleEntryBlueprintMapping` plus the number of `ArgumentDataPrototype`s aggregated in the role argument of the `clientServerOperation`

⌋*()*

[constr_2598] `ClientServerOperationBlueprintMapping` constrains the types of arguments ⌈The arguments in the ordered lists `bswModuleEntry` and the matching arguments in the set union of the ordered lists `portDefinedArgumentBlueprint` plus `clientServerOperation` shall result in the identical C data type definitions.

⌋*()*

[constr_2603] Use of "applies to" in context of the specification level ⌈On specification level 1 and 2 only the requirements table including the appliesTo attribute shall be used. On the specification levels 3 and 4 only the requirements table without the appliesTo attribute shall be used. Exception: Documents of the foundation which are handled on specification level 3.

⌋*()*

[constr_2604] Allowed uptraces in context of "applies to" values ⌈Traces to documents of upper specification levels shall be conform to the values assigned to appliesTo.

⌋*()*

[constr_2608] Custom extensions shall be part of the `Documentation` that is referenced by the `Baseline` ⌈If a `SpecElementReference` references a custom defined specification element, then this specification element shall be part of a `Documentation` that is referenced by the `Baseline` of this `Profile`.

⌋*()*

[constr_2609] Single revision per AUTOSAR standard ⌈
The `standardRevision` may only contain a single revision per AUTOSAR standard. E.g. it is allowed to combine the AUTOSAR standards "Foundation" in revision 1.0.0 with the "Classic Platform" in revision 4.3.0. However, it is not allowed to reference the revisions 4.2.2 and 4.3.0 of the "Classic Platform" in the same `Baseline`.

⌋*()*

[constr_2610] No alternativeName if matching via shortName ⌈The `alternativeName` shall not be set if the referenced AUTOSAR Specification Element matches the rules of `Identifier`.

⌋*()*

[constr_2611] Referenced AUTOSAR Specification Elements shall be part of the AUTOSAR Specification Baseline ⌈If the `SpecElementReference` references an AUTOSAR specification element then the `shortName` or `alternativeName` shall match the name of the AUTOSAR specification element in a specification that is part of the revision of the standard that is specified in `Baseline`.

⌋*()*

[constr_2612] `shortName` of `ConcreteClassTailoring` shall match the name of an AUTOSAR specified concrete meta-class ⌈`shortName` of `ConcreteClassTailoring` shall match the name of an AUTOSAR specified concrete meta-class).

⌋*()*

[constr_2613] `shortName` of `AbstractClassTailoring` shall match the name of an AUTOSAR specified abstract meta-class ⌈`shortName` of `AbstractClassTailoring` shall match the name of an AUTOSAR specified abstract meta-class).

⌋*()*

[constr_2614] `PrimitiveAttributeCondition.attribute` shall reference invariant owned `PrimitiveAttributeTailoring`, only ⌈The following conditions need to evaluate to true for `PrimitiveAttributeCondition.attribute`:

- The referenced `PrimitiveAttributeTailoring` is owned by an `ClassContentConditional` that has no `condition` (invariant class content) **AND**

- The `ClassContentConditional` that owns the referenced `PrimitiveAttributeTailoring` and the `ClassContentConditional` that owns this `PrimitiveAttributeCondition` are owned by the same `ClassTailoring`.

⌋*()*

[constr_2615] `AggregationCondition.aggregation` shall reference invariant owned `AggregationTailoring`, only ⌈The following conditions need to evaluate to true for `AggregationCondition.aggregation`:

- The referenced `AggregationTailoring` is owned by an `ClassContentConditional` that has no `condition` (invariant class content) **AND**

- The `ClassContentConditional` that owns the referenced `AggregationTailoring` and the `ClassContentConditional` that owns this `AggregationCondition` are owned by the same `ClassTailoring`.

⌋*()*

[constr_2616] `ReferenceCondition.reference` shall reference invariant owned `ReferenceTailoring`, only ⌈The following conditions need to evaluate to true for `ReferenceCondition.reference`:

- The referenced `ReferenceTailoring` is owned by an `ClassContentConditional` that has no `condition` (invariant class content) **AND**

- The `ClassContentConditional` that owns the referenced `ReferenceTailoring` and the `ClassContentConditional` that owns this `ReferenceCondition` are owned by the same `ClassTailoring`.

⌋*()*

[constr_2617] `ClassTailoring.variationRestriction` only applicable for <<atpVariation>> classes ⌈If the tailored meta class is not marked with stereotype <<atpVariation>> then `ClassTailoring.variationRestriction` shall not be defined.

⌋*()*

[constr_2618] ShortName of AttributeTailoring shall match owned or inherited attributes ⌈The `shortName` shall match the name of an attribute that is owned or inherited by the AUTOSAR meta-class which is identified by the `ClassTailoring` that owns this `AttributeTailoring`.

⌋*()*

[constr_2619] No `AttributeTailoring` for Derived or Abstract Attributes ⌈No `AttributeTailoring`s are allowed for `Attributes` that are marked with stereotypes <<atpDerived>> or <<atpAbstract>>.

⌋*()*

[constr_2620] `shortName` of `PrimitiveAttributeTailoring` shall be a primitive attribute in the referenced Baseline ⌈The `shortName` of `PrimitiveAt-`

`tributeTailoring` shall match the name of an AUTOSAR specified primitive attribute of the Meta-Class in the referenced Baseline.

⌋*()*

[constr_2621] The `shortName` of `AggregationTailoring` shall match the name of an AUTOSAR specified aggregation of the meta-class ⌈The `shortName` of `AggregationTailoring` shall match the name of an AUTOSAR specified aggregation of the meta-class).

⌋*()*

[constr_2622] The `shortName` of `ReferenceTailoring` shall match the name of an AUTOSAR specified reference of the meta-class ⌈The `shortName` of `ReferenceTailoring` shall match the name of an AUTOSAR specified reference of the meta-class).

⌋*()*

[constr_2623] Referenced `SdgClass` shall be part of a `SdgDef` that is referenced by the `Baseline` ⌈Referenced `SdgClass` shall be part of a `SdgDef` that is referenced by the `Baseline` of this `Profile of Data Exchange Point`.

⌋*()*

[constr_2624] `AttributeTailoring.variationRestriction` only applicable for <<atpVariation>> attributes ⌈If the tailored attribute is not marked with stereotype <<atpVariation>> then `AttributeTailoring.variationRestriction` shall not be defined.

⌋*()*

[constr_2625] Allowed uptraces wrt. life cycles ⌈

Trace from	Trace to			
	draft	valid	obsolete	removed
draft	1	1	0	0
valid	0	1	0	0
obsolete	1	1	1	0
removed	1	1	1	1

⌋*()*

2.8   TPS_SystemTemplate

[constr_1001] Value of `dataId` shall be unique ⌈The value of the `dataId` shall be unique within the scope of the `System`.

⌋*()*

[constr_1002] End-to-end protection does not support n:1 communication ⌈As the n:1 communication scenario implies that probably not all senders use the same `dataId` this scenario is explicitly not supported.

⌋*()*

[constr_1198] `TriggerToSignalMapping.systemSignal`s eligible for a `TriggerToSignalMapping` in case no `DataTransformation` is used ⌈The `ISignal` that is referenced by a `SystemSignal` that in turn is referenced by a `TriggerToSignalMapping` in the role `systemSignal` shall have the length attribute set to 0 if the `ISignal` does not reference a `DataTransformation` in the role `dataTransformation`.

⌋*()*

[constr_1199] `ISignal`s relating to `systemSignal`s eligible for a `TriggerToSignalMapping` shall use update bit in case no `DataTransformation` is used ⌈An `ISignal`

- that is used to reference a `systemSignal` that in turn is referenced by a `TriggerToSignalMapping` and

- does not reference a `DataTransformation` in the role `dataTransformation`

shall be referenced by an `ISignalToIPduMapping` where the attribute `updateIndicationBitPosition` is defined.

⌋*()*

[constr_1265] `DoIpGidSynchronizationNeeds` can only exist once per `ECU_EXTRACT` ⌈Within the context of one `System` of `category` ECU_EXTRACT, there can only be at most one `DoIpGidSynchronizationNeeds`.

⌋*()*

[constr_1266] `DoIpGidNeeds` can only exist once per `ECU_EXTRACT` ⌈Within the context of one `System` of `category` ECU_EXTRACT, there can only be at most one `DoIpGidNeeds`.

⌋*()*

[constr_1267] `DoIpActivationLineNeeds` can only exist once per `ECU_EXTRACT` ⌈Within the context of one `System` of `category` ECU_EXTRACT, there can only be at most one `DoIpActivationLineNeeds`.

⌋*()*

[constr_1367] `periodicResponseUudt.periodicResponseUudt` shall only refer to a `DcmIPdu` ⌈If the role `periodicResponseUudt` exists then every `PduTriggering` referenced in the role `periodicResponseUudt` shall only refer to a `DcmIPdu`.

⌋*()*

[constr_1368] Limitation of the target of references from `DiagnosticConnection` ⌈`DiagnosticConnection` shall only reference (via the indirection created by `TpConnectionIdent`) the following sub-classes of the meta-class `TpConnection`:

- `CanTpConnection`
- `FlexrayTpConnection`
- `FlexrayArTpConnection`
- `DoIpTpConnection`

⌋*()*

[constr_1369] `CommunicationConnector`s shall be attached to the same `CommunicationCluster` ⌈All `CommunicationConnector`s referenced from `GlobalTimeMaster` and `GlobalTimeSlave`s aggregated in one `GlobalTimeDomain` shall be referenced in the role `commConnector` by the same `PhysicalChannel` aggregated by the same `CommunicationCluster`.

⌋*()*

[constr_1370] Consistency of `GlobalTimeDomain` ⌈The `GlobalTimeSlave` referenced in the role `GlobalTimeGateway.slave` and the `GlobalTimeMaster` referenced in the role `GlobalTimeGateway.master` shall **not** be aggregated by the same `GlobalTimeDomain`.

⌋*()*

[constr_1371] Consistency of attribute `host` ⌈Within the context of an aggregating `GlobalTimeDomain`, the `CommunicationConnector`s referenced in the role `GlobalTimeGateway.master.communicationConnector` and `GlobalTimeGateway.slave.communicationConnector` shall be aggregated by the same `EcuInstance` that is referenced in the role `GlobalTimeGateway.host`.

⌋*()*

[constr_1372] Consistency of attribute `pduTriggering` ⌈Within the context of an aggregating `GlobalTimeDomain`, the `pduTriggering` shall be owned by `PhysicalChannel` that is also referencing the `CommunicationConnector`s referenced in the roles `GlobalTimeSlave.communicationConnector` and `GlobalTimeMaster.communicationConnector`.

⌋*()*

[constr_1373] `GlobalTimeMaster` with attribute `isSystemWideGlobalTimeMaster` set to `TRUE` ⌈`GlobalTimeMaster` with attribute `isSystemWideGlobalTimeMaster` set to `TRUE` shall not be referenced in the role `GlobalTimeGateway.master`.

⌋*()*

[constr_1374] Only fan-out possible for `GlobalTimeGateway` ⌈For all `Global-TimeGateway`s that refer to the same `EcuInstance` the condition applies that no two `GlobalTimeGateway`s shall refer to the same `GlobalTimeMaster`.

⌋*()*

[constr_1387] Transmission of `Variable-Size Array Data Type`s by means of a Transformer ⌈If a Transformer is used for the transmission of a `Variable-Size Array Data Type`s then the `Variable-Size Array Data Type` shall be a "new-world" variable-size array data type according to [TPS_SWCT_01644] and [TPS_-SWCT_01645]. "Old-world" dynamic-size array data types according to [TPS_SWCT_-01641] and [TPS_SWCT_01642] are not supported.

⌋*()*

[constr_1441] In AUTOSAR, the transmission of union data types over the network is only supported by the SOME/IP Transformer ⌈If an `Implementation-DataType` according to [TPS_SWCT_01700], i.e. of `category` STRUCT that encloses an `ImplementationDataTypeElement` of `category` UNION, is used to directly or (via a `DataTypeMap`) indirectly type an `AutosarDataPrototype` and the latter is mapped to a `SystemSignal` then the `ISignal` that references that `SystemSignal` shall aggregate `SOMEIPTransformationISignalProps` in the role `transformationISignalProps`.

⌋*()*

[constr_1463] Applicable values for `J1939Cluster.networkId` ⌈The values of the attribute `J1939Cluster.networkId` shall always be within the interval 1..4.

⌋*()*

[constr_1641] Consistent assignment of TLV data ids to `ApplicationRecord-DataType` ⌈For every `ApplicationRecordDataType` where direct members set the attribute `ApplicationRecordElement.isOptional` to the value `True` references to all direct members of this `ApplicationRecordDataType` shall be created on the basis of the definition of `TlvDataIdDefinition`.

⌋*()*

[constr_1642] Consistent assignment of TLV data ids to `Implementation-DataType` or `ImplementationDataTypeElement` ⌈For every `Implementation-tionDataType` or `ImplementationDataTypeElement` of `category` STRUCTURE where direct members set the attribute `ImplementationDataTypeElement.isOptional` to the value `True` references to all direct members of this `Implementation-DataType` resp `ImplementationDataTypeElement` shall be created on the basis of the definition of `TlvDataIdDefinition`.

⌋*()*

[constr_1643] Completeness of the existence of a set of `TlvDataIdDefinition.tlvArgument`s ⌈If the reference `TlvDataIdDefinition.tlvArgument`s exists for one `argument` of a given `ClientServerOperation` then further `TlvDataIdDef-`

`inition`.`tlvArgument`s shall exist for all `argument`s of the given `ClientServer-Operation` and all affected `TlvDataIdDefinition`s shall be referenced by the same `SOMEIPTransformationISignalProps` via `TlvDataIdDefinitionSet`.

⌋*()*

[constr_1644] Completeness of the existence of a set of `TlvDataIdDefinition.tlvRecordElement`s ⌈If the reference `TlvDataIdDefinition`.`tlvRecordElement` exists for one element of a given `ApplicationRecordDataType` then further `TlvDataIdDefinition`.`tlvRecordElement` shall exist for all elements of the given `ApplicationRecordDataType` and all affected `TlvDataIdDefinition`s shall be referenced by the same `SOMEIPTransformationISignalProps` via `TlvDataIdDefinitionSet`.

⌋*()*

[constr_1645] Completeness of the existence of a set of `TlvDataIdDefinition.tlvImplementationDataTypeElement`s ⌈Completeness of the existence of a set of `TlvDataIdDefinition`.`tlvImplementationDataTypeElement`s If the reference `TlvDataIdDefinition`.`tlvImplementationDataTypeElement` exists for one `subElement` of a given `ImplementationDataType` or `ImplementationDataTypeElement` then further `TlvDataIdDefinition`.`tlvImplementationDataTypeElement` shall exist for all `subElement`s of the given `ImplementationDataType` or `ImplementationDataTypeElement` and all affected `TlvDataIdDefinition`s shall be referenced by the same `SOMEIPTransformationISignalProps` via `TlvDataIdDefinitionSet`.

⌋*()*

[constr_1646] Scope of the uniqueness of the value of `TlvDataIdDefinition.id` for references to `ArgumentDataPrototype` ⌈For all `TlvDataIdDefinition` that are referencing `ArgumentDataPrototype`s of a given `ClientServerOperation` in the role `tlvArgument` the attribute `TlvDataIdDefinition`.`id` shall exist and have a unique value in the context of respective `argument`s of the enclosing `ClientServerOperation` where attribute `direction` is set to the value `in`/`inout` or `out`/`inout`.

Note: an `argument` where attribute `direction` is set to the value `in` may have the same data id as an `argument` where attribute `direction` is set to the value `out` since the two are transferred in separate messages.

⌋*()*

[constr_1647] Scope of the uniqueness of the value of `TlvDataIdDefinition.id` for references to `ApplicationRecordElement` ⌈For all `TlvDataIdDefinition` that are referencing `ApplicationRecordElement`s of a given `ApplicationDataType` in the role `tlvRecordElement` the attribute `TlvDataIdDefini-`

`tion`.id shall exist and have a unique value in the context of respective enclosing `ApplicationRecordDataType`.

⌋*()*

[constr_1648] Scope of the uniqueness of the value of `TlvDataIdDefinition`.id for references to `ImplementationDataTypeElement` ⌈For all `TlvDataIdDefinition` that are referencing `ImplementationDataTypeElement`s of a given `ImplementationDataType`/`ImplementationDataTypeElement` in the role `tlvImplementationDataTypeElement` the attribute `TlvDataIdDefinition`.id shall exist and have a unique value in the context of respective enclosing `ImplementationDataType` or `ImplementationDataTypeElement`.

⌋*()*

[constr_1649] `TlvDataIdDefinition` referencing `ArgumentDataPrototype` ⌈Each `ArgumentDataPrototype` shall be referenced at most once in the role `tlvArgument` in the context of the same `SOMEIPTransformationISignalProps`.

⌋*()*

[constr_1650] `TlvDataIdDefinition` referencing `ApplicationRecordElement` ⌈Each `ApplicationRecordElement` shall be referenced at most once in the role `tlvRecordElement` in the context of the same `SOMEIPTransformationISignalProps`.

⌋*()*

[constr_1651] `TlvDataIdDefinition` referencing `ImplementationDataTypeElement` ⌈Each `ImplementationDataTypeElement` shall be referenced at most once in the role `tlvImplementationDataTypeElement` in the context of the same `SOMEIPTransformationISignalProps`.

⌋*()*

[constr_1652] Definition of static length fields sizes in case of TLV usage ⌈If `TlvDataIdDefinition`s are defined for a `SOMEIPTransformationISignalProps`, the attributes `sizeOfArrayLengthFields`, `sizeOfStructLengthFields`, `sizeOfStringLengthFields` and `sizeOfUnionLengthFields` shall be greater than 0.

⌋*()*

[constr_1653] Identical values for length fields sizes in case of TLV usage ⌈If `TlvDataIdDefinition`s are defined for a `SOMEIPTransformationISignalProps`, the attributes `sizeOfArrayLengthFields`, `sizeOfStructLengthFields`, `sizeOfStringLengthFields` and `sizeOfUnionLengthFields` shall have an identical value.

⌋*()*

[constr_1654] No definition of length field sizes on `DataPrototype` level in case of TLV usage ⌈If `TlvDataIdDefinition`s are defined for a `SOMEIPTransformationISignalProps`, the attributes `sizeOfArrayLengthFields`, `sizeOfStructLengthFields` and `sizeOfUnionLengthFields` shall not be defined on `DataPrototype` level but only on `ISignal` level.

⌋*()*

[constr_1655] The mutual existence of `LinMaster`s in the `LinSlave` EcuExtract ⌈A `LinMaster` shall not be part of the EcuExtract of a corresponding `LinSlave`.

⌋*()*

[constr_1656] No application-level write access to `LinErrorResponse.responseError` on Lin slave ⌈The `SystemSignal` referenced in the role `systemSignal` by the `ISignal` referenced by the `ISignalTriggering` that in turn is referenced in the role `LinErrorResponse.responseError` shall not be referenced by a `DataMapping` that allows for writing to the `SystemSignal`.

⌋*()*

[constr_1657] Existence of `LinPhysicalChannel.scheduleTable` ⌈In any given Ecu Extract that contains a `LinSlave`, the `LinPhysicalChannel` that relates to the respective `LinSlave` via `commConnector`.`commController` shall not aggregate a `LinScheduleTable`.

⌋*()*

[constr_1669] Existence of `PduTriggering.secOcCryptoMapping` ⌈The reference `PduTriggering`.`secOcCryptoMapping` shall only exist if the `PduTriggering` also references a `SecuredIPdu` in the role `iPdu`.

⌋*()*

[constr_1670] Prohibition of usage of `tlsCryptoMapping` in case of UDP socket connections ⌈A `TlsCryptoServiceMapping` may only be referenced by an `ApplicationEndpoint` in the role `tlsCryptoMapping` if that `ApplicationEndpoint` aggregates a `TcpTp` in the role `tpConfiguration`.

⌋*()*

[constr_1671] Supported values of `TlsCryptoServiceMapping.category` ⌈The only supported values of attribute `TlsCryptoServiceMapping`.`category` are:

- **`TLS_SERVER`**: the `TlsCryptoServiceMapping` assumes the role of the *server* in the TLS connection.

- **`TLS_CLIENT`**: the `TlsCryptoServiceMapping` assumes the role of the *client* in the TLS connection.

⌋*()*

[constr_1672] **Existence of `TlsCryptoCipherSuite.certificate` and `TlsCryptoCipherSuite.pskIdentity` in the *server* role** ⌈Either

- the reference to `CryptoServiceCertificate` in the role `TlsCryptoCipherSuite.certificate`

- the aggregation of `TlsPskIdentity` in the role `TlsCryptoCipherSuite.pskIdentity`

shall exist if the `TlsCryptoCipherSuite` is aggregated by a `TlsCryptoServiceMapping` that has attribute `category` set to the value `TLS_SERVER`.

⌋*()*

[constr_2025] **Uniqueness of `symbol` attributes** ⌈With the exception of RunnableEntities that are subject to [constr_1234] (RunnableEntities owned by NvBlockSwComponentTypes), in the context of a single `EcuInstance` the values of the `RunnableEntity.symbol` in combination with the attribute `symbol` of the meta-class `SymbolProps` owned by `AtomicSwComponentType` of all deployed RunnableEntities shall be unique such that no two (or more) combinations of `RunnableEntity.symbol` and the `symbol` of the meta-class `SymbolProps` owned by `AtomicSwComponentType` in the role `symbolProps` share the same value.

⌋*()*

[constr_3000] **valid `SenderRecCompositeTypeMapping`s** ⌈All `SenderRecRecordElementMapping`s or `SenderRecArrayElementMapping`s aggregated in the context of a given `SenderReceiverToSignalGroupMapping` shall reference a `SystemSignal` that is also referenced in the role `systemSignal` by the `SystemSignalGroup` that is referenced by the enclosing `SenderReceiverToSignalGroupMapping` in the role `signalGroup`.

⌋*()*

[constr_3002] **valid swcToImplMapping** ⌈The referenced `SwcImplementation` refers to a `SwcInternalBehavior` that is part of a `AtomicSwComponentType`. The same `AtomicSwComponentType` shall be the type of the referenced `SwComponentPrototype`.

SwcToImplMapping.componentImplementation.behavior.component == SwcToImplMapping.component.type

⌋*()*

[constr_3003] **Number of CAN channels** ⌈CAN clusters shall aggregate exactly one `PhysicalChannel`.

⌋*()*

[constr_3004] **Clustering and separation shall be exclusive** ⌈Clustering and separation shall be exclusive, i.e. it SHALL NOT be possible that two `SwComponent-`

`Prototype`s A and B are associated both by a `ComponentClustering` and by a `ComponentSeparation` at the same time.

⌋*()*

[constr_3005] valid `EcuResourceEstimation` ⌈The same `EcuInstance` shall be referenced directly from the `EcuResourceEstimation` and from the `SwcToEcuMapping`:

EcuResourceEstimation.swCompToEcuMapping.ecuInstance == EcuResourceEstimation.ecuInstance

⌋*()*

[constr_3006] valid EcuMapping ⌈The referenced `hwCommunicationController` and `hwCommunicationPort` shall be part of the referenced `ecu`.

`ECUMapping`.`ecu`.nestedElement contains `ECUMapping`.`commControllerMapping`.`hwCommunicationController`

`ECUMapping`.`ecu`.nestedElement contains `ECUMapping`.`hwPortMapping`.`hwCommunicationPort`

⌋*()*

[constr_3007] `selectorFieldCode`s for dynamic part alternatives ⌈The `selectorFieldCode`s for the dynamic part alternatives within one `MultiplexedIPdu` shall differ from each other.

⌋*()*

[constr_3008] `EcuInstance` subelements ⌈The `CommunicationConnector` and the `CommunicationController` that is referenced by the `CommunicationConnector` shall be owned by the same `EcuInstance`.

⌋*()*

[constr_3009] Overlapping of `ISignal`s is prohibited ⌈`ISignal`s mapped to an `ISignalIPdu` shall not overlap.

⌋*()*

[constr_3010] `ISignalIPdu` length shall not be exceeded ⌈The combined length of all `ISignal`s and `updateIndicationBitPosition`s that are mapped into an `ISignalIPdu` shall not exceed the defined `Pdu length`.

⌋*()*

[constr_3011] Overlapping of `updateIndicationBits` of `ISignal`s is prohibited ⌈The `updateIndicationBitPosition` for an `ISignal` in an `ISignalIPdu`

shall not overlap with other `updateIndicationBitPosition`s or `ISignal` locations.
⌋*()*

[constr_3012]   Overlapping of `Pdu`s is prohibited ⌈`Pdu`s mapped to a `FlexrayFrame` shall NOT overlap.
⌋*()*

[constr_3013] `FlexrayFrame` length shall not be exceeded ⌈The combined length of all `Pdu`s that are mapped into a `FlexrayFrame` shall not exceed the defined `FlexrayFrame` length.
⌋*()*

[constr_3014] Overlapping of updateIndicationBits for `Pdu`s is prohibited ⌈The `updateIndicationBitPosition` for a `Pdu` in a `FlexrayFrame` shall NOT overlap with other `updateIndicationBitPosition`s and `Pdu` locations.
⌋*()*

[constr_3015] Number of LIN channels ⌈LIN clusters shall aggregate exactly one `LinPhysicalChannel`.
⌋*()*

[constr_3018]   Number of FlexRay channels ⌈A `FlexrayCluster` shall use either one `FlexrayPhysicalChannel` with `channelName` set to either `channelA` or `channelB` or else two `FlexrayPhysicalChannel`s with one `channelName channelA` and one `channelName channelB`.
⌋*()*

[constr_3019] In the flat ECU extract each required interface shall be satisfied by connected provided interfaces ⌈In case of the flat `System` with `category` ECU_EXTRACT all `VariableDataPrototype`s specified by the `SenderReceiverInterface` of the `RPortPrototype` need to be supplied by some of the `PPortPrototype`s being connected with `SwConnector`s.
⌋*()*

[constr_3020] `communicationDirection` of `containedISignalIPduGroup`s ⌈The value of the attribute `communicationDirection` of `containedISignalIPduGroup` shall be identical to the value of the attribute `communicationDirection` of the enclosing `ISignalIPduGroup`.
⌋*()*

[constr_3021] Mapping of `SensorActuatorSwComponent`s to SensorActuator `HwElement`s ⌈Only `SwComponentPrototype`s that are typed by `SensorActua-`

`torSwComponentType` shall be mapped to a `HwElement` with `category` Sensor Actuator via the `controlledHwElement` relation.

⌋*()*

[constr_3025] Usage of `NPdu`s in `TpConnections` ⌈In case several `TpConnec-tion`s use the same Frame ID for their communication needs only one `NPdu` element per Frame Id shall exist. This constraint applies for all supported AUTOSAR transport protocols (CanTp, LinTp, FrTp, FrArTp and J1939Tp).

⌋*()*

[constr_3027] Existence of `ecuExtractVersion` ⌈In case the category of the System is SYSTEM_EXTRACT or ECU_EXTRACT the `ecuExtractVersion` attribute shall be defined.

⌋*()*

[constr_3028] `FibexElements` ⌈Each `FibexElement` that is used in the System Description shall be referenced by the `System` element in the role `FibexElement`.

⌋*()*

[constr_3029] Assign-Frame command usage ⌈For the LIN 2.0 Assign-Frame command the `LinConfigurableFrame` list shall be used. For the LIN 2.1 Assign-Frame-PID-Range command the `LinOrderedConfigurableFrame` list shall be used.

⌋*()*

[constr_3030] valid relationship between `ECUMapping` and `EcuInstance` ⌈If an `EcuInstance` is assigned to a `HwElement` the `EcuInstance` shall belong to the same `System` as the `ECUMapping`.

⌋*()*

[constr_3031] Complete System Description does not have ports on the outermost composition ⌈In a complete `System` with `category` ABSTRACT_SYSTEM_DE-SCRIPTION or `System` with `category` SYSTEM_DESCRIPTION this outermost `CompositionSwComponentType` has the unique feature that it doesn't have any outside ports, but all the SWC contained in it are connected to each other and fully specified by their `SwComponentType`s, `PortPrototype`s, `PortInterface`s, `VariableDataPrototype`s, `InternalBehavior` etc.

⌋*()*

[constr_3036] `Pdu`s in CAN and LIN Frames ⌈CAN Frames and LIN Frames shall only contain one `Pdu`.

⌋*()*

[constr_3037] maximum `Frame frameLength` for CAN and LIN ⌈For CAN and LIN the maximum `frameLength` is 8 bytes and 64 bytes in case of CAN FD.

⌋*()*

[constr_3038] maximum `Frame frameLength` for FlexRay ⌈For FlexRay the maximum `frameLength` is 254 bytes.

⌋*()*

[constr_3039] `pncIdentifier` range ⌈The `pncIdentifier` value shall be in the range of 8..63 for normal CAN and in the range of 8..511 for CAN FD, FlexRay and Ethernet.

⌋*()*

[constr_3040] Restriction of `pncIdentifier` values ⌈The `pncIdentifier` value shall be within the range described by `pncVectorOffset` and `pncVectorLength`.

⌋*()*

[constr_3044] CBV configuration in case partial network is used ⌈In case a partial network is used the control bit vector (CBV) shall be defined in Byte 0 of the `NmPdu` (`nmCbvPosition` = 0).

⌋*()*

[constr_3045] Signal content evaluation vs. Mode evaluation ⌈The mode evaluation and the signal content evaluation shall not be used in the same `IPdu`. A mix of these two types is not allowed.

⌋*()*

[constr_3046] Consistency of `TransmissionModeCondition.iSignalInIPdu` ⌈The `ISignalToIPduMapping` referenced by the `TransmissionModeCondition` in the role `iSignalInIPdu` shall belong to the same `ISignalIPdu` as the `TransmissionModeCondition`.

⌋*()*

[constr_3047] Uniqueness of `macMulticastAddress`es ⌈A `macMulticastAddress` shall be unique in a particular `EthernetCluster`.

⌋*()*

[constr_3048] Range of `vlanIdentifier` ⌈The allowed values of `vlanIdentifier` range from 0 to 4095.

⌋*()*

[constr_3050] `J1939Cluster` uses exactly one `CanPhysicalChannel` ⌈A `J1939Cluster` shall aggregate exactly one `CanPhysicalChannel`.

⌋*()*

[constr_3051] Restriction of `ISignalMapping` references ⌈If the `sourceSignal` references an `ISignal` then the `targetSignal` shall also reference an `ISignal`.

⌋*()*

[constr_3052] Complete `ISignalMapping` of `ISignalGroup` signals ⌈If an `ISignalMapping` to an `ISignal` that is a member of a `ISignalGroup` exists then (see [TPS_SYST_01120]) an `ISignalMapping` to the enclosing `ISignalGroup` shall exist as well.

⌋*()*

[constr_3053] Complete `ISignalMapping` of target `ISignalGroup` ⌈If an `ISignalGroup` is referenced by a `targetSignal` then [TPS_SYST_02162] applies for each of the contained `ISignal` of that `ISignalGroup`.

⌋*()*

[constr_3057] Maximal one `BusspecificNmEcu` per `NmEcu` and bus system is allowed to be defined ⌈For each `NmEcu` at most one `BusspecificNmEcu` per bus system (FlexRay/Can/Udp/J1939) is allowed to be defined.

⌋*()*

[constr_3058] References from `SenderRecArrayElementMapping` and from `SenderRecRecordElementMapping` to `SystemSignal`s are not allowed within a `SenderReceiverCompositeElementToSignalMapping` ⌈The reference from `SenderRecArrayElementMapping` to `SystemSignal` and from `SenderRecRecordElementMapping` to `SystemSignal` shall not exist if the enclosing `SenderRecCompositeTypeMapping` is owned by a `SenderReceiverCompositeElementToSignalMapping`.

⌋*()*

[constr_3060] Allowed Attributes for `networkRepresentationProps` and `physicalProps` ⌈

Attributes of SwDataDefProps	SystemSignal.physicalProps	ISignal.networkProps
`additionalNativeTypeQualifier`	NA	NA
`annotation`	NA	NA
`baseType`	NA	D
`baseType.category`	NA	M
`BaseTypeDirectDefinition.baseTypeEncoding`	NA	D
`BaseTypeDirectDefinition.byteOrder`	NA	NA
`BaseTypeDirectDefinition.baseTypeSize`	NA	0..1
`BaseTypeDirectDefinition.memAlignment`	NA	NA
`BaseTypeDirectDefinition.nativeDeclaration`	NA	NA
`compuMethod`	D	I
`dataConstr`	D	M
`displayFormat`	D	M
`implementationDataType`	NA	NA

▽

△

invalidValue	NA	D
stepSize	NA	NA
swAddrMethod	NA	NA
swAlignment	NA	NA
swBitRepresentation	NA	NA
swCalibrationAccess	NA	NA
swCalprmAxisSet	NA	NA
swComparisonVariable	NA	NA
swDataDependency	NA	NA
swHostVariable	NA	NA
swImplPolicy	NA	NA
swIntendedResolution	NA	NA
swInterpolationMethod	NA	NA
swIsVirtual	NA	NA
swPointerTargetProps	NA	NA
swRecordLayout	NA	NA
swRefreshTiming	NA	NA
swTextProps	NA	NA
swValueBlockSize	NA	NA
unit	D	M
valueAxisDataType	NA	NA

⌋*()*

[constr_3062] The `EcuInstance` that is referenced from a specific `CouplingElement` shall be connected to the same `EthernetCluster` as the specific `CouplingElement` ⌈The `EcuInstance` referenced from a specific `CouplingElement` in the role `ecuInstance` shall be connected via the `CommunicationConnector` and a `EthernetPhysicalChannel` that refers the `CommunicationConnector` to the `EthernetCluster` referenced by the specific `CouplingElement` in the role `communicationCluster`.

⌋*()*

[constr_3067] `initValue` defined in the context of `ISignal` ⌈The definition of an `initValue` in the context of an `ISignal` shall only be a `NumericalValueSpecification`, `TextValueSpecification` or `ArrayValueSpecification` that aggregates elements of type `NumericalValueSpecification` or `TextValueSpecification`.

⌋*()*

[constr_3068] `DoIpPowerModeStatusNeeds` in the `category` ECU_EXTRACT ⌈If and only if DoIP (i.e. any of the subclasses of `DoIpServiceNeeds` are present) is used on an Ecu then the `DoIpPowerModeStatusNeeds` shall exist exactly once in a `System` of category ECU_EXTRACT.

⌋*()*

[constr_3069] Allowed `CanNmCluster.nmNidPosition` values ⌈If defined, the value of `CanNmCluster.nmNidPosition` shall only be set to either 0 or 1.

⌋*()*

[constr_3070] Allowed `CanNmCluster.nmCbvPosition` values ⌈If defined, the value of `CanNmCluster.nmCbvPosition` shall only be set to either 0 or 1.

⌋*()*

[constr_3071] `CanNmCluster.nmCbvPosition` and `CanNmCluster.nmNidPosition` shall never have the same value ⌈`CanNmCluster.nmCbvPosition` and `CanNmCluster.nmNidPosition` shall never have the same value.

⌋*()*

[constr_3073] `nmVoteInformation` only valid for FrNm ⌈The `nmVoteInformation` attribute is only valid for FrNm.

⌋*()*

[constr_3074] No TransmissionAcknowledgementRequest for multiple senders ⌈If more than one `SenderComSpec` exist (in different `PortPrototype`s on atomic level) that refer to data elements effectively mapped to the same `SystemSignal` it is not allowed that any `SenderComSpec` aggregates `transmissionAcknowledge`.

⌋*()*

[constr_3078] Allowed `UdpNmCluster.nmNidPosition` values ⌈If defined, the value of `UdpNmCluster.nmNidPosition` shall only be set to either 0 or 1.

⌋*()*

[constr_3079] Allowed `UdpNmCluster.nmCbvPosition` values ⌈If defined, the value of `UdpNmCluster.nmCbvPosition` shall only be set to either 0 or 1.

⌋*()*

[constr_3080] `UdpNmCluster.nmCbvPosition` and `UdpNmCluster.nmNidPosition` shall never have the same value ⌈`UdpNmCluster.nmCbvPosition` and `UdpNmCluster.nmNidPosition` shall never have the same value.

⌋*()*

[constr_3081] Value of category in `GeneralPurposePdu` ⌈The attribute `category` of `GeneralPurposePdu` can have the following values:

- SD (Service Discovery)
- GLOBAL_TIME
- DoIP

⌋*()*

[constr_3082] Value of category in `GeneralPurposeIPdu` ⌈The attribute `category` of `GeneralPurposeIPdu` can have the following values:

- XCP

- SOMEIP_SEGMENTED_IPDU

- DLT

⌋*()*

[constr_3083] Exactly one `AtomicSwComponentType` on an `EcuInstance` may use `GeneralCallbackEventDataChanged` / `GeneralCallbackEventStatusChange` ⌈The Dem only supports exactly one `AtomicSwComponentType` using `GeneralCallbackEventDataChanged` / `GeneralCallbackEventStatusChange` on one `EcuInstance`.

⌋*()*

[constr_3084] Service port in the role PowerTakeOff ⌈Within the context of one `EcuInstance`, there can only be one service port that uses the role PowerTakeOff in the `RoleBasedPortAssignment.role`.

⌋*()*

[constr_3085] Service port in the role CallbackDCMRequestServices ⌈Within the context of one `EcuInstance`, there can only be one service port that uses the role CallbackDCMRequestServices in the `RoleBasedPortAssignment.role`.

⌋*()*

[constr_3086] Role of `SystemSignal` in n:1 sender-receiver communication ⌈In case of n:1 communications

- if `DataTransformation` is used each sender shall be mapped to the same `SystemSignal`

- if `DataTransformation` is not used each sender shall be mapped

    - to the same `SystemSignal` in case of a primitive DataType on the sender side,

    - to the same `SystemSignalGroup` in case of a composite DataType on the sender side.

⌋*()*

[constr_3090] `TpSdu` transmission on a `PhysicalChannel` ⌈The `IPdu` that is referenced by a `TpConnection` in the role `tpSdu` shall be referenced by exactly one `PduTriggering` aggregated on the `PhysicalChannel` of the `TpConnection`.

⌋*()*

[constr_3094] Consistent `ISignalPort.communicationDirection` for `ISignalTriggering`s of `ISignalGroup`s and contained `ISignal`s ⌈In case the

`ISignal`s contained in an `ISignalGroup` are referenced by an `ISignalTriggering`, the `communicationDirection` of the `ISignalPort` referenced by the `ISignal`'s `ISignalTriggering` shall be identical to the `communicationDirection` of the `ISignalPort` referenced by the containing `ISignalGroup`'s `ISignalTriggering`.

⌋*()*

[constr_3095] canControllerFdAttributes and canControllerFdRequirements are mutually exclusive ⌈The existence of `canControllerFdAttributes` and `canControllerFdRequirements` is mutually exclusive.

⌋*()*

[constr_3096] Allowed values for `diagnosticMessageType` ⌈The allowed values of `diagnosticMessageType` range from 1..57.

⌋*()*

[constr_3097] Overlapping of segments of one `MultiplexedIPdu` is not allowed ⌈The segments defined by the `SegmentPosition` elements of one and the same `MultiplexedIPdu` - aggregated via `StaticPart` and `DynamicPart` - shall not overlap.

⌋*()*

[constr_3098] Defined segments of one `MultiplexedIPdu` shall not exceed the length of the `MultiplexedIPdu` ⌈The segments defined by the `SegmentPosition` elements of one and the same `MultiplexedIPdu` - aggregated via `StaticPart` and `DynamicPart` - shall not exceed the length of the `MultiplexedIPdu`.

⌋*()*

[constr_3099] Defined segments in a `DynamicPart` shall not exceed the length of any `DynamicPartAlternative.iPdu` ⌈The segments defined by the `SegmentPosition` elements aggregated in the `DynamicPart` of a `MultiplexedIPdu` shall not exceed the length of any `DynamicPartAlternative.iPdu`.

⌋*()*

[constr_3100] Defined segments in a `StaticPart` shall not exceed the length of the `StaticPart.iPdu` ⌈The segments defined by the `SegmentPosition` elements aggregated in the `StaticPart` of a `MultiplexedIPdu` shall not exceed the length of the `StaticPart.iPdu`

⌋*()*

[constr_3101] Signal representation of selector field for `DynamicPartAlternative` ⌈Every `ISignalIPdu` that is referenced by the `DynamicPartAlternative` shall contain an `ISignal` that represents the selector field. The selector field signal

shall be located at the position that is described by the `selectorFieldLength` and `selectorFieldStartPosition`.

⌋*()*

[constr_3102] **Restriction on usage of `J1939NodeName` attributes** ⌈A `J1939NmCluster` shall not aggregate two `J1939NmNode`s with identical `J1939NodeName` attributes.

⌋*()*

[constr_3103] Range of `ecuInstance` ⌈The allowed values of `ecuInstance` range from 0 to 7.

⌋*()*

[constr_3104] Range of `function` ⌈The allowed values of `function` range from 0 to 255.

⌋*()*

[constr_3105] Range of `functionInstance` ⌈The allowed values of `functionInstance` range from 0 to 31.

⌋*()*

[constr_3106] **Range of `identitiyNumber`** ⌈The allowed values of `identitiyNumber` range from 0 to 2097151.

⌋*()*

[constr_3107] Range of `industryGroup` ⌈The allowed values of `industryGroup` range from 0 to 7.

⌋*()*

[constr_3108] Range of `manufacturerCode` ⌈The allowed values of `manufacturerCode` range from 0 to 2047.

⌋*()*

[constr_3109] Range of `vehicleSystem` ⌈The allowed values of `vehicleSystem` range from 0 to 127.

⌋*()*

[constr_3110] Range of `vehicleSystemInstance` ⌈The allowed values of `vehicleSystemInstance` range from 0 to 15.

⌋*()*

[constr_3111] `returnSignal` in `ClientServerToSignalMapping` is mandatory ⌈A `ClientServerToSignalMapping` shall always have a `returnSignal` defined.

⌋*()*

[constr_3112] Invalidation support for partial mapping of a data element typed by composite data type ⌈If a `VariableDataPrototype` with a composite data type in a `PPortPrototype` is mapped to a `SystemSignalGroup` and only a subset of elements of the composite data type that are primitives is mapped to separate `SystemSignal`s of the `SystemSignalGroup` then at least one mapped primitive shall have an `invalidValue` defined.

⌋*()*

[constr_3113] `AbstractEthernetFrame` shall not have a `PduToFrameMapping` ⌈It is not allowed to map `Pdu`s into `AbstractEthernetFrame`s.

⌋*()*

[constr_3114] `FlatInstanceDescriptor`s pointing to the same `ParameterDataPrototype` shall have different `postBuildVariantCondition`s ⌈`FlatInstanceDescriptor`s that are pointing as an atpTarget to the same `ParameterDataPrototype` instance shall have different `postBuildVariantCondition`s.

⌋*()*

[constr_3115] `FlatInstanceDescriptor`s pointing to the same `ParameterDataPrototype` instance ⌈When several `FlatInstanceDescriptor`s point to the same `ParameterDataPrototype` instance as an atpTarget in the context of a `ParameterInterface` the different `FlatInstanceDescriptor`s shall point to the `PPortPrototype` of the owning `ParameterSwComponentType`. In this case the `PPortPrototype` typed by the `ParameterInterface` is part of the context of the according `AnyInstanceRef`.

⌋*()*

[constr_3116] Overlap of `ClientIdRange`s in the context of the enclosing System ⌈The `ClientIdRange` defined for an `EcuInstance` shall not overlap with the `ClientIdRange` of any other `EcuInstance` in the context of the enclosing System.

⌋*()*

[constr_3117] Allowed value of attribute `clientId` ⌈Within the context of one `ClientIdDefinition`, the value of attribute `clientId` shall be in the range of `ClientIdRange.lowerLimit` and `ClientIdRange.upperLimit` for the `ClientIdRange` that is aggregated by the `EcuInstance` onto which the `SwComponentPrototype`s included in the `ClientIdDefinition.clientServerOperation` are mapped.

⌋*()*

[constr_3118] Valid reference target for `ClientIdDefinition.clientServerOperation.contextPort` ⌈In the context of the definition of a `ClientIdDefinition`, the reference `clientServerOperation.contextPort` shall only refer to an `RPortPrototype`.

⌋*()*

[constr_3121] The length of transformer chains is limited to 255 transformers ⌈The maximum number of `DataTransformation.transformerChain` references in the context of one `DataTransformation` shall be limited to 255.

⌋*()*

[constr_3122] At most one transformer of each transformer class inside a transformer chain ⌈If the value of a `transformerClass` of a `TransformationTechnology` referenced by a `DataTransformation` does not equal `custom`, it shall be different from all other `transformerClass` values of `TransformationTechnology`s referenced by the same `DataTransformation`.

⌋*()*

[constr_3123] Serializer transformer shall be the first in a chain ⌈A serializer transformer (`TransformationTechnology` with attribute `transformerClass` set to `serializer`) shall be the first transformer in a transformer chain.

⌋*()*

[constr_3124] Applicability of `needsOriginalData` ⌈The attribute `needsOriginalData` of a `TransformationTechnology` shall only be used for the non-first transformers in the transformer chain.

⌋*()*

[constr_3125] Value of attribute `inPlace` for the first transformer in a chain ⌈The attribute `inPlace` shall be set to `false` if the `TransformationTechnology` of the `BufferProperties` is referenced as first reference in the ordered list of references `transformerChain` from a `DataTransformation`.

⌋*()*

[constr_3127] Certain `ISignal`s always need a reference to `DataTransformation` ⌈An `ISignal` which references a `SystemSignal` which is referenced by a `SystemSignalGroup` in the role `transformingSystemSignal` shall always reference a `DataTransformation`.

⌋*()*

[constr_3128] SOME/IP transformer configuration ⌈For each `TransformationDescription` variant that is a `SOMEIPTransformationDescription`

- attribute `protocol` of `TransformationTechnology` shall be set to `SOMEIP`
- attribute `version` of `TransformationTechnology` shall be set to `1.0.0`
- attribute `transformerClass` of `TransformationTechnology` shall be set to `serializer`
- attribute `headerLength` of `BufferProperties` shall be set to `64` (bits).

⌋*()*

[constr_3129] Byte Order of SOME/IP transformer ⌈The attribute `byteOrder` of `SOMEIPTransformationDescription` shall be different from `opaque`.

⌋*()*

[constr_3130] Range of Interface Version ⌈The value of the attribute `interfaceVersion` shall be in the range $[0; 255]$

⌋*()*

[constr_3132] Required COM Based Transformation for `comBasedSignalGroupTransformation` ⌈If a `ISignalGroup` has a reference to the `DataTransformation` element in the role `comBasedSignalGroupTransformation` then this `DataTransformation` shall be the handled by the COM Based Transformer [11].

⌋*()*

[constr_3133] `physicalLayerType` of connected `CouplingPort`s ⌈The `physicalLayerType` of two `CouplingPort`s which are connected via a `CouplingPortConnection` shall be equal.

⌋*()*

[constr_3134] The connection of two `CouplingPort`s with `connectionNegotiationBehavior` set to `master` is forbidden ⌈The `connectionNegotiationBehavior` of two `CouplingPort`s which are connected via a `CouplingPortConnection` shall not be both set to `master`.

⌋*()*

[constr_3135] The connection of two `CouplingPort`s with `connectionNegotiationBehavior` set to `slave` is forbidden ⌈The `connectionNegotiationBehavior` of two `CouplingPort`s which are connected via a `CouplingPortConnection` shall not be both set to `slave`.

⌋*()*

[constr_3136] Allowed payload of `SecuredIPdu`s ⌈`SecuredIPdu`s are allowed to reference `PduTriggering`s of `ISignalIPdu`s, `ContainerIPdu`s, `DcmIPdu`s, `MultiplexedIPdu`s, `GeneralPurposeIPdu`s with `category` SOMEIP_SEGMENTED_IPDU and `UserDefinedIPdu`s.

⌋*()*

[constr_3137] `IPduPort.rxSecurityVerification` is configurable on the receiver side ⌈The `IPduPort.rxSecurityVerification` attribute shall only be used in `IPduPort`s with the `communicationDirection` = in.

⌋*()*

[constr_3138] `IPduPort.rxSecurityVerification` validness ⌈The `IPduPort.rxSecurityVerification` information is only valid for `SecuredIPdu`s.

⌋*()*

[constr_3140] **No `ByteOrderEnum.opaque` allowed for `System.container-IPduHeaderByteOrder`** ⌈The values of `System.containerIPduHeaderByte-Order` are restricted to `ByteOrderEnum.mostSignificantByteFirst` and `ByteOrderEnum.mostSignificantByteLast`. I.e. the value `ByteOrderEnum.opaque` is not allowed.

⌋*()*

[constr_3141] **Only `IPdu`s shall be part of a `ContainerIPdu`** ⌈The `PduTriggering` which is referenced in the role `ContainerIPdu.containedPduTriggering` or `ContainerIPdu.containedIPduTriggeringProps.containedPduTriggering` shall refer to a subclass of an `IPdu` in the role `PduTriggering.iPdu`.

⌋*()*

[constr_3142] **Mandatory `headerIdLongHeader` for `longHeader`** ⌈For each `IPdu` which is assigned to a `ContainerIPdu` in the role `ContainerIPdu.containedPduTriggering` or `ContainerIPdu.containedIPduTriggeringProps.containedPduTriggering` with `ContainerIPdu.headerType` = `longHeader` the `ContainedIPduProps.headerIdLongHeader` shall be defined.

⌋*()*

[constr_3143] **Mandatory `headerIdShortHeader` for `shortHeader`** ⌈For each `IPdu` which is assigned to a `ContainerIPdu` in the role `ContainerIPdu.containedPduTriggering` or `ContainerIPdu.containedIPduTriggeringProps.containedPduTriggering` with `ContainerIPdu.headerType` = `shortHeader` the `ContainedIPduProps.headerIdShortHeader` shall be defined.

⌋*()*

[constr_3144] Mandatory `IPdu.containedIPduProps` for contained `IPdu`s ⌈For each `IPdu` which is assigned to a `ContainerIPdu` in the role `ContainerIPdu.containedPduTriggering` the `IPdu.containedIPduProps` shall be defined.

⌋*()*

[constr_3146] Partial Networking timing constraint ⌈For Partial Networking the following timing constraints shall be ensured:

- CAN / Ethernet: (`pnResetTime` + `pncPrepareSleepTimer`) < `nmNetworkTimeout`
- FlexRay: (`pnResetTime` + `pncPrepareSleepTimer`) < `nmReadySleepTime`

⌋*()*

[constr_3148] `executeDespiteDataUnavailability` setting in case an E2E Transformer is used ⌈A transformer chain using E2E shall be configured with `DataTransformation.executeDespiteDataUnavailability` = TRUE.

⌋*()*

[constr_3149] `TransformationTechnology.needsOriginalData` **settings for E2E Transformer** ⌈The `TransformationTechnology.needsOriginalData` attribute of a `TransformationTechnology` element of an E2E transformer shall be set to FALSE.

⌋*()*

[constr_3151] `BufferProperties.headerLength` **settings for an E2E transformer used in combination with a SOME/IP transformer** ⌈The `BufferProperties.headerLength` for an E2E transformer located in a transformer chain with a SOME/IP transformer shall be configured with the following values depending on the value of the `EndToEndTransformationDescription.profileName` attribute:

1. PROFILE_01: `BufferProperties.headerLength` = 16 bits

2. PROFILE_02: `BufferProperties.headerLength` = 16 bits

3. PROFILE_04: `BufferProperties.headerLength` = 96 bits

4. PROFILE_05: `BufferProperties.headerLength` = 24 bits

5. PROFILE_06: `BufferProperties.headerLength` = 40 bits

6. PROFILE_07: `BufferProperties.headerLength` = 160 bits

7. PROFILE_08: `BufferProperties.headerLength` = 128 bits

8. PROFILE_11: `BufferProperties.headerLength` = 16 bits

9. PROFILE_22: `BufferProperties.headerLength` = 16 bits

10. PROFILE_04m: `BufferProperties.headerLength` = 128 bits

11. PROFILE_07m: `BufferProperties.headerLength` = 192 bits

12. PROFILE_08m: `BufferProperties.headerLength` = 160 bits

13. PROFILE_44: `BufferProperties.headerLength` = 96 bits

14. PROFILE_44m: `BufferProperties.headerLength` = 128 bits

⌋*()*

[constr_3152] `BufferProperties.headerLength` **settings for any transformer used in combination with a COM Based transformer** ⌈A transformer used in a transformer chain with a COM Based transformer shall be configured with the following values:

- `BufferProperties.headerLength` = 0

⌋*()*

[constr_3153] E2E header field reservation required by COM Based transformer ⌈A COM Based transformer that is used in a transformer chain with an E2E transformer requires that the following amount of space is allocated for the E2E header fields using a proper `ISignalGroup` layout according to [TPS_SYST_02068]:

PROFILE_01: if `dataIdMode` == `lower12Bit`: 16 bits

PROFILE_01: if `dataIdMode` != `lower12Bit`: 12 bits

PROFILE_02: 16 bits

PROFILE_04: 96 bits

PROFILE_05: 24 bits

PROFILE_06: 40 bits

PROFILE_07: 160 bits

PROFILE_08: 128 bits

PROFILE_11: if `dataIdMode` == `lower12Bit`: 16 bits

PROFILE_11: if `dataIdMode` == `all16Bit`: 12 bits

PROFILE_22: 16 bits

PROFILE_04m: 128 bits

PROFILE_07m: 192 bits

PROFILE_08m: 160 bits

PROFILE_44: 96 bits

PROFILE_44m: 128 bits

⌋*()*

[constr_3155] Allowed values for `EndToEndTransformationDescription.upperHeaderBitsToShift` ⌈The value of of the `EndToEndTransformationDescription.upperHeaderBitsToShift` attribute depends on the used serializing transformer:

COM based transformer: 0 (no bits are shifted)

SOME/IP transformer: 64 (to support the header shift of SOME/IP).

Custom transformer: no restriction (depends on header length and placement of custom transformer)

⌋*()*

[constr_3156] Allowed values for `EndToEndTransformationISignalProps.dataId` in PROFILE_01 and PROFILE_11 ⌈If the `EndToEndTransformationDescription.profileName` attribute has a value of PROFILE_01 or PROFILE_11 then the value of the `EndToEndTransformationISignalProps.dataId` attribute shall be in the range of 0-65535.

⌋*()*

[constr_3157] **Allowed values for `EndToEndTransformationISignalProps.`** **`dataId`** **in PROFILE_01 and PROFILE_11 in case `dataIdMode` is set to** **`lower12Bit`** ⌈If the `EndToEndTransformationDescription.profileName` attribute has a value of PROFILE_01 or PROFILE_11 and the value of `EndToEndTransformationDescription.dataIdMode` attribute has a value of `lower12Bit` then the value of the `EndToEndTransformationISignalProps.dataId` attribute shall be in the range of 256-65535.

⌋*()*

[constr_3158] **Allowed values for `EndToEndTransformationDescription.`** **`maxDeltaCounter` in PROFILE_01 and PROFILE_11** ⌈If the `EndToEndTransformationDescription.profileName` attribute has a value of PROFILE_01 or PROFILE_11 then the attribute `maxDeltaCounter` shall be in the range 1-14.

⌋*()*

[constr_3159] **Allowed values for `EndToEndTransformationDescription.`** **`maxDeltaCounter` in PROFILE_04, PROFILE_04m PROFILE_44 and PROFILE_44m** ⌈If the `EndToEndTransformationDescription.profileName` attribute has a value of PROFILE_04, PROFILE_04m, PROFILE_44, or PROFILE_44m the value of `maxDeltaCounter` attribute shall be in the range 1-65535.

⌋*()*

[constr_3160] **`EndToEndTransformationISignalProps.dataId` in PROFILE_02 and PROFILE_22** ⌈If the `EndToEndTransformationDescription.profileName` attribute has a value of PROFILE_02 or PROFILE_22 then the multiplicity of the `dataId` attribute shall be 16 and the value of each instance shall be in the range 0..255.

⌋*()*

[constr_3161] **`EndToEndTransformationISignalProps.dataLength` in PROFILE_01, PROFILE_02, PROFILE_05, PROFILE_11, PROFILE_22** ⌈If the `EndToEndTransformationDescription.profileName` attribute has a value of PROFILE_01, PROFILE_02, PROFILE_05, PROFILE_11, or PROFILE_22 then the multiplicity of the `EndToEndTransformationISignalProps.dataLength` attribute shall be 1.

⌋*()*

[constr_3162] **`EndToEndTransformationISignalProps.minDataLength` and** **`EndToEndTransformationISignalProps.maxDataLength` in PROFILE_01, PROFILE_02, PROFILE_05, PROFILE_11, PROFILE_22** ⌈If the `EndToEndTransformationDescription.profileName` attribute has a value of PROFILE_01, PROFILE_02, PROFILE_05, PROFILE_11, or PROFILE_22 then the multiplicity of the attributes `EndToEndTransformationISignalProps.minDataLength` and `EndToEndTransformationISignalProps.maxDataLength` shall be 0.

⌋*()*

[constr_3163] `EndToEndTransformationISignalProps.minDataLength` **and** `EndToEndTransformationISignalProps.maxDataLength` **in PROFILE_04, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, and PROFILE_44m** ⌈If the `EndToEndTransformationDescription.profileName` attribute has a value of PROFILE_04, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, or PROFILE_44m then the multiplicity of the attributes `EndToEndTransformationISignalProps.minDataLength` and `EndToEndTransformationISignalProps.maxDataLength` shall be 1.

⌋*()*

[constr_3164] `EndToEndTransformationISignalProps.dataLength` **in PROFILE_04, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44 and PROFILE_44m** ⌈If the `EndToEndTransformationDescription.profileName` attribute has a value of PROFILE_04, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, or PROFILE_44m then the multiplicity of the attribute `EndToEndTransformationISignalProps.dataLength` shall be 0.

⌋*()*

[constr_3165] **Effect of** `EndToEndTransformationDescription.upperHeaderBitsToShift` **value in PROFILE_01, PROFILE_11** ⌈If the `EndToEndTransformationDescription.profileName` attribute has a value of PROFILE_01 or PROFILE_11 and the serializing transformer is different than the ComBasedTransformer then:

1. `EndToEndTransformationDescription.crcOffset` shall be set to the same value of `upperHeaderBitsToShift`.

2. `EndToEndTransformationDescription.counterOffset` shall be set to the value of `upperHeaderBitsToShift` + 8.

3. (if used) `EndToEndTransformationDescription.dataIdNibbleOffset` shall be set to the value of `upperHeaderBitsToShift` + 12.

⌋*()*

[constr_3166] `EndToEndTransformationDescription.upperHeaderBitsToShift` **in PROFILE_02** ⌈If the `EndToEndTransformationDescription.profileName` attribute has a value of PROFILE_02 then the value of the `upperHeaderBitsToShift` attribute shall be 0.

⌋*()*

[constr_3167] **Effect of** `EndToEndTransformationDescription.upperHeaderBitsToShift` **value in PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, and PROFILE_44m** ⌈If the `EndToEndTransformationDescription.profileName` attribute has a value of PROFILE_04, PROFILE_05, PROFILE_06,

PROFILE_07, PROFILE_08, PROFILE_04m, PROFILE_07m, PROFILE_08m, PRO-FILE_44, or PROFILE_44m the value of the `EndToEndTransformationDescription.offset` attribute shall be equal to the value of the `EndToEndTransformationDescription.upperHeaderBitsToShift` attribute.

⌋*()*

[constr_3169] `EndToEndTransformationDescription.offset` value in PRO-FILE_02 and PROFILE_22 ⌈If the `EndToEndTransformationDescription.profileName` attribute has a value of PROFILE_02 or PROFILE_22 then the value of the `EndToEndTransformationDescription.offset` attribute shall be 0.

⌋*()*

[constr_3172] Effect of `EndToEndTransformationDescription.profileBehavior` value in PROFILE_01 ⌈If the `EndToEndTransformationDescription.profileName` attribute has a value of PROFILE_01 and the value of the `profileBehavior` attribute is `R4_2` then:

- the value of the `EndToEndTransformationDescription.maxNoNewOrRepeatedData` attribute shall be 14.

- the value of the `EndToEndTransformationDescription.syncCounterInit` attribute shall be 1.

⌋*()*

[constr_3173] Effect of `EndToEndTransformationDescription.profileBehavior` value in PROFILE_02 ⌈If the `EndToEndTransformationDescription.profileName` attribute has a value of PROFILE_02 and the value of the `profileBehavior` attribute is `R4_2` then:

- the value of the `EndToEndTransformationDescription.maxNoNewOrRepeatedData` attribute shall be 15.

- the value of the `EndToEndTransformationDescription.syncCounterInit` attribute shall be 1.

⌋*()*

[constr_3174] `EndToEndTransformationDescription` settings not allowed in PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PRO-FILE_11, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PRO-FILE_44 and PROFILE_44m ⌈If the `EndToEndTransformationDescription.profileName` attribute has a value of PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_11, PROFILE_22, PROFILE_04m, PRO-FILE_07m, PROFILE_08m, PROFILE_44 or PROFILE_44m then:

1. the multiplicity of the `EndToEndTransformationDescription.maxNoNewOrRepeatedData` attribute shall be 0.

2. the multiplicity of the `EndToEndTransformationDescription`.`syncCounterInit` attribute shall be 0.

3. the multiplicity of the `EndToEndTransformationDescription`.`profileBehavior` attribute shall be 0.

⌋*()*

[constr_3182] Restriction on `TransformationTechnology`.`transformationDescription` `VariationPoint` ⌈The `EndToEndTransformationDescription`.`profileName` attribute shall not be subject to variability for a given `ISignal` / `ISignalGroup`, i.e., the value of the `EndToEndTransformationDescription`.`profileName` attribute shall be the same in all different variants.

⌋*()*

[constr_3183] `ISignalGroup` with `transformationISignalProps` ⌈An `ISignalGroup` that aggregates `transformationISignalProps` shall reference the `DataTransformation` in the role `comBasedSignalGroupTransformation`.

⌋*()*

[constr_3184] Only one `EndToEndTransformationISignalProps`.`dataId` element in PROFILE_01 and PROFILE_11 ⌈If the `EndToEndTransformationDescription`.`profileName` attribute has a value of PROFILE_01 or PROFILE_11 then the multiplicity of the `EndToEndTransformationISignalProps`.`dataId` attribute shall be 1.

⌋*()*

[constr_3185] Multiplicity of `EndToEndTransformationDescription`.`dataIdMode` in PROFILE_01 and PROFILE_11 ⌈If the `EndToEndTransformationDescription`.`profileName` attribute is set to PROFILE_01 or PROFILE_11 then the multiplicity of the `EndToEndTransformationDescription`.`dataIdMode` attribute shall be 1.

⌋*()*

[constr_3186] Multiplicity of `EndToEndTransformationDescription`.`dataIdMode` in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44 and PROFILE_44m ⌈If the `EndToEndTransformationDescription`.`profileName` attribute is set to a value of PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, or PROFILE_44m then the multiplicity of the `EndToEndTransformationDescription`.`dataIdMode` attribute shall be 0.

⌋*()*

[constr_3187] Multiplicity of `EndToEndTransformationDescription`.`counterOffset` in PROFILE_01 and PROFILE_11 ⌈If the `EndToEndTransformationDescription`.`profileName` attribute is set to PROFILE_01 or PROFILE_11 then

the multiplicity of the `EndToEndTransformationDescription.counterOffset` attribute shall be 1.

⌋*()*

[constr_3188] Multiplicity of `EndToEndTransformationDescription.counterOffset` in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, and PROFILE_44m ⌈If the `EndToEndTransformationDescription.profileName` attribute is set to a value of PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, or PROFILE_44m then the multiplicity of the `EndToEndTransformationDescription.counterOffset` attribute shall be 0.

⌋*()*

[constr_3189] Multiplicity of `EndToEndTransformationDescription.crcOffset` in PROFILE_01 and PROFILE_11 ⌈If the `EndToEndTransformationDescription.profileName` attribute is set to PROFILE_01 or PROFILE_11 then the multiplicity of the `EndToEndTransformationDescription.crcOffset` attribute shall be 1.

⌋*()*

[constr_3190] Multiplicity of `EndToEndTransformationDescription.crcOffset` in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44 and PROFILE_44m ⌈If the `EndToEndTransformationDescription.profileName` attribute is set to a value of PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, or PROFILE_44m then the multiplicity of the `EndToEndTransformationDescription.crcOffset` attribute shall be 0.

⌋*()*

[constr_3191] Multiplicity of `EndToEndTransformationDescription.dataIdNibbleOffset` in PROFILE_01, PROFILE_11 and `dataIdMode` equal to `lower12Bit` ⌈If the `EndToEndTransformationDescription.profileName` attribute is set to PROFILE_01 or PROFILE_11 and the value of the `EndToEndTransformationDescription.dataIdMode` attribute is set to `lower12Bit` then the multiplicity of the `EndToEndTransformationDescription.dataIdNibbleOffset` attribute shall be 1.

⌋*()*

[constr_3192] Multiplicity of `EndToEndTransformationDescription.dataIdNibbleOffset` in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, and PROFILE_44m or `dataIdMode` different from `lower12Bit` ⌈If the `EndToEndTransformationDescription.profileName` at-

tribute is set to a value of PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44, or PROFILE_44m or the `EndToEndTransformationDescription.dataIdMode` attribute is set to value different from `lower12Bit` then the multiplicity of the `EndToEndTransformationDescription.dataIdNibbleOffset` attribute shall be 0.

⌋()

[constr_3193] Multiplicity of `EndToEndTransformationDescription.offset` in PROFILE_01 and PROFILE_11 ⌈If the `EndToEndTransformationDescription.profileName` attribute is set to PROFILE_01 or PROFILE_11 then the multiplicity of the `EndToEndTransformationDescription.offset` attribute shall be 0.

⌋()

[constr_3194] Multiplicity of `EndToEndTransformationDescription.offset` in PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44 and PROFILE_44m ⌈If the `EndToEndTransformationDescription.profileName` attribute is set to a value PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_08, PROFILE_22, PROFILE_04m, PROFILE_07m, PROFILE_08m, PROFILE_44 or PROFILE_44m then the multiplicity of the `EndToEndTransformationDescription.offset` attribute shall be 1.

⌋()

[constr_3195] Allowed values for `EndToEndTransformationDescription.maxDeltaCounter` in PROFILE_02 and PROFILE_22 ⌈If the `EndToEndTransformationDescription.profileName` attribute has a value of PROFILE_02 or PROFILE_22 then the attribute `maxDeltaCounter` shall be in the range 1-15.

⌋()

[constr_3196] Allowed values for `EndToEndTransformationDescription.maxDeltaCounter` in PROFILE_05 ⌈If the `EndToEndTransformationDescription.profileName` attribute has a value of PROFILE_05 then the attribute `maxDeltaCounter` shall be in the range 1-255.

⌋()

[constr_3197] Allowed values for `EndToEndTransformationDescription.maxDeltaCounter` in PROFILE_06 ⌈If the `EndToEndTransformationDescription.profileName` attribute has a value of PROFILE_06 then the attribute `maxDeltaCounter` shall be in the range 1-255.

⌋()

[constr_3198] Uniqueness of `PncMapping.shortLabel` ⌈If the optional `shortLabel` attribute is used it shall be unique in the `System` scope.

⌋*()*

[constr_3199] `ISignal` that has `dataTypePolicy` set to `transformingISignal` shall reference a `DataTransformation` ⌈In a complete model every `ISignal` that has `dataTypePolicy` set to `transformingISignal` shall reference a `DataTransformation`.

⌋*()*

[constr_3202] `LinFrameTriggering` to `LinUnconditionalFrame` reference restriction in `LinEventTriggeredFrame` context ⌈Within a `PhysicalChannel` a `LinUnconditionalFrame` shall be referenced by only one `LinFrameTriggering` to allow a derivation of the identifier of a substituted Frame if the `LinUnconditionalFrame` is referenced by a `LinEventTriggeredFrame` in the role `linUnconditionalFrame`.

⌋*()*

[constr_3203] `LinFrameTriggering` to `LinSporadicFrame` reference restriction in `LinSporadicFrame` context ⌈Within a `PhysicalChannel` a `LinUnconditionalFrame` shall be referenced by only one `LinFrameTriggering` to allow a derivation of the identifier of a substituted Frame if the `LinUnconditionalFrame` is referenced by a `LinSporadicFrame` in the role `substitutedFrame`.

⌋*()*

[constr_3204] `LinUnconditionalFrame`s associated with a `LinSporadicFrame` ⌈A `LinUnconditionalFrame` associated with a `LinSporadicFrame` shall not be allocated in the same `LinScheduleTable` as the `LinSporadicFrame`.

⌋*()*

[constr_3205] Existence of `FramePort` for a `FrameTriggering` that references a `LinSporadicFrame` ⌈A `FrameTriggering` that references a `LinSporadicFrame` shall not have a reference to a `FramePort`.

⌋*()*

[constr_3206] Existence of `FramePort` for a `FrameTriggering` that references a `LinEventTriggeredFrame` ⌈A `FrameTriggering` that references a `LinEventTriggeredFrame` shall not have a reference to a `FramePort`.

⌋*()*

[constr_3208] `executeDespiteDataUnavailability` usage restriction ⌈In the set of more than one `ISignal` which reference the same `SystemSignal` in the role `systemSignal`, there shall be no `ISignal` which references a `DataTransformation` where `executeDespiteDataUnavailability` is set to true.

⌋*()*

[constr_3209] **CanFrameTriggering**s **with identical PGN** ⌈For all `Can-FrameTriggering`s where the attribute `identifier` contains the identical PGN (as defined in section 5.2 Protocol Data Unit in [12]) the attribute `j1939requestable` shall also have an identical value.

⌋*()*

[constr_3210] **J1939TpPg**s **with identical** **pgn** **value** ⌈For all `J1939TpPg`s where the attribute `pgn` has an identical value the attribute `requestable` shall also have an identical value.

⌋*()*

[constr_3211] **PduTriggering**s **with** **triggerIPduSendCondition** ⌈Only `PduTriggering`s with references to `ISignalIPdu`s are allowed to contain a `triggerIPduSendCondition`.

⌋*()*

[constr_3212] Limitation of DoIpTpConnection.tpSdu ⌈`DoIpTpConnection` shall only reference `PduTriggering`s of `DcmIPdu`s or `UserDefinedIPdu`s in the role `tpSdu`.

⌋*()*

[constr_3213] **TransformationISignalProps.csErrorReaction** **setting in case that the** **serializer transformerClass** **and Client/Server communication is used** ⌈In `TransformationISignalProps` the attribute `csErrorReaction` shall be set if the `TransformationISignalProps` specifies the details for a `TransformationTechnology` with `transformerClass` equal to `serializer` and the `ISignal` that aggregates the `TransformationISignalProps` transports a client/ server communication.

⌋*()*

[constr_3214] **TransformationISignalProps.csErrorReaction** **setting in case that a** **transformerClass** **different from** **serializer** **is used or the Client/ Server communication is not used** ⌈In `TransformationISignalProps` the attribute `csErrorReaction` shall not be used if the `TransformationISignalProps` specifies the details for a `TransformationTechnology` with `transformerClass` not equal to `serializer` or the `ISignal` that aggregates the `Transformation-ISignalProps` does not transport a client/server communication.

⌋*()*

[constr_3215] **TransformationTechnology.version** **and** **Transformation-Technology.protocol** **settings for request and response of a client/server communication** ⌈`TransformationTechnology`.version and `Transformation-Technology`.protocol shall be identical for `ISignal`s that are derived from the same `ClientServerOperation`. This means that all `ISignal`s that refer to `ClientServerToSignalMapping`.callSignal or to `ClientServerToSignalMapping`.returnSignal of the same `ClientServerToSignalMapping` shall

have the same `TransformationTechnology`.`protocol` and `Transformation-Technology`.`version` defined.

⌋*()*

[constr_3216] Usage of `SOMEIPTransformationISignalProps`.`sessionHandlingSR` ⌈The attribute `sessionHandlingSR` of `SOMEIPTransformationISignalProps` shall only be used for `ISignal`s which reference `SystemSignal`s which are mapped via a `SenderReceiverToSignalMapping`.

⌋*()*

[constr_3218] Range of Size of Array Length Fields ⌈The value of attribute `sizeOfArrayLengthFields` of `SOMEIPTransformationISignalProps` shall be either 0, 1, 2 or 4.

⌋*()*

[constr_3219] The mutual existence of `LinSlave`s in the `LinMaster` EcuExtract ⌈`LinSlave`s shall not be part of the EcuExtract of the corresponding `LinMaster`.

⌋*()*

[constr_3220] Range of Size of Structure Length Fields ⌈The value of attribute `sizeOfStructLengthFields` of `SOMEIPTransformationISignalProps` shall be either 0, 1, 2 or 4.

⌋*()*

[constr_3221] Range of Size of Union Length Fields ⌈The value of attribute `sizeOfUnionLengthFields` of `SOMEIPTransformationISignalProps` shall be either 0, 1, 2 or 4.

⌋*()*

[constr_3222] No `ByteOrderEnum`.`opaque` allowed for `PduToFrameMapping`.`packingByteOrder` ⌈The values of `PduToFrameMapping`.`packingByteOrder` are restricted to `ByteOrderEnum`.`mostSignificantByteFirst` and `ByteOrderEnum`.`mostSignificantByteLast`. I.e. the value `ByteOrderEnum`.`opaque` is not allowed.

⌋*()*

[constr_3223] No `ByteOrderEnum`.`opaque` allowed for `MultiplexedIPdu`.`selectorFieldByteOrder` ⌈The values of `MultiplexedIPdu`.`selectorFieldByteOrder` are restricted to `ByteOrderEnum`.`mostSignificantByteFirst` and `ByteOrderEnum`.`mostSignificantByteLast`. I.e. the value `ByteOrderEnum`.`opaque` is not allowed.

⌋*()*

[constr_3224] No `ByteOrderEnum`.`opaque` allowed for `SegmentPosition`.`segmentByteOrder`. ⌈The values of `SegmentPosition`.`segmentByteOrder` are re-

stricted to `ByteOrderEnum`.`mostSignificantByteFirst` and `ByteOrderEnum`.`mostSignificantByteLast`. I.e. the value `ByteOrderEnum`.`opaque` is not allowed.

⌋*()*

[constr_3225] **`LinFrameTriggering`.`linChecksum` not allowed for `LinSporadicFrame`s** ⌈The `linChecksum` attribute of a `LinFrameTriggering` that references a `LinSporadicFrame` shall not be set.

⌋*()*

[constr_3226] **`LinFrameTriggering`.`linChecksum` for `LinEventTriggeredFrame`s** ⌈Within a `PhysicalChannel` the `linChecksum` attribute of a `LinFrameTriggering` that references a `LinEventTriggeredFrame` shall have the same value as the `linChecksum` attribute of each `LinFrameTriggering` that references a `LinUnconditionalFrame` that in turn is referenced by that `LinEventTriggeredFrame`.

⌋*()*

[constr_3227] **`NmNode`.`nmPassiveModeEnabled` setting** ⌈`NmNode`.`nmPassiveModeEnabled` shall be set to the same value in all `NmCluster`s with the same bus protocol in the scope of one `NmEcu`.

⌋*()*

[constr_3229] **`SwComponentPrototype` mapped to an `ApplicationPartition` and `EcuInstance`** ⌈If the `SwcToEcuMapping`.`ecuInstance` exists then a `SwComponentPrototype` that is mapped to an `ApplicationPartition` via the `SwcToApplicationPartitionMapping` shall only be mapped by an `ApplicationPartitionToEcuPartitionMapping` to an `EcuPartition` that is aggregated by the `EcuInstance` referenced by means of `SwcToEcuMapping`.`ecuInstance`.

⌋*()*

[constr_3230] **Usage of `SenderRecRecordElementMapping`.`applicationRecordElement`** ⌈`SenderRecRecordElementMapping`.`applicationRecordElement` shall only be used if the referenced context element (`VariableDataPrototype` that is referenced by the `SenderReceiverToSignalGroupMapping`.`dataElement`) is typed by an `ApplicationDataType`.

⌋*()*

[constr_3231] **Usage of `IndexedArrayElement`.`applicationArrayElement`** ⌈`IndexedArrayElement`.`applicationArrayElement` shall only be used if the referenced context element (`VariableDataPrototype` that is referenced by the `SenderReceiverToSignalGroupMapping`.`dataElement`) is typed by an `ApplicationDataType`.

⌋*()*

[constr_3232] `ApplicationPartition` is allowed to be mapped to only one `EcuPartition` ⌈Each `ApplicationPartition` shall be mapped at most once to an `EcuPartition` via the `ApplicationPartitionToEcuPartitionMapping`.

⌋*()*

[constr_3239] Consistent mapping of software-component to `J1939NmNode` ⌈The value of attribute `J1939NmNode`.`nodeName`.`function` of a `J1939NmNode` referenced by `J1939ControllerApplicationToJ1939NmNodeMapping` in the role `j1939NmNode` shall be identical to the value of `J1939ControllerApplication`.`functionId`.

⌋*()*

[constr_3240] Consistent mapping of `J1939ControllerApplication` to `EcuInstance` ⌈A `SwComponentPrototype` that is referenced by a `J1939ControllerApplication` mapped to a specific `J1939NmNode` shall only be mapped to an `EcuInstance` that in turn owns the same `J1939NmNode`.

⌋*()*

[constr_3243] `FrameTriggering.pduTriggering` condition ⌈A `FrameTriggering` shall reference a `PduTriggering` if the `PduTriggering` references a `Pdu` that is referenced by a `PduToFrameMapping` which in turn is aggregated by the `Frame` that is referenced by that `FrameTriggering`.

⌋*()*

[constr_3244] Usage of `SenderRecRecordElementMapping.implementationRecordElement` ⌈`SenderRecRecordElementMapping`.`implementationRecordElement` shall only be used if the referenced context element (`VariableDataPrototype` that is referenced by the `SenderReceiverToSignalGroupMapping`.`dataElement`) is typed by an `ImplementationDataType`.

⌋*()*

[constr_3245] Usage of `IndexedArrayElement.implementationArrayElement` ⌈`IndexedArrayElement`.`implementationArrayElement` shall only be used if the referenced context element (`VariableDataPrototype` that is referenced by the `SenderReceiverToSignalGroupMapping`.`dataElement`) is typed by an `ImplementationDataType`.

⌋*()*

[constr_3246] `Frame.packingByteOrder` mix within a `Frame` is not allowed ⌈All `PduToFrameMapping`s within a `Frame` shall have the same `packingByteOrder` value.

⌋*()*

[constr_3247] Byte order mix within a `MultiplexedIPdu` is not allowed ⌈The `segmentByteOrder` of all `SegmentPosition`s and the `selectorFieldByteOrder` shall have the same value in the `MultiplexedIPdu`.

⌋*()*

[constr_3248] Category of `HwElement` for `ECUMapping` ⌈The `HwElement` which is referenced from `ECUMapping` in the role `ecu` shall be of category `MicroController`

⌋*()*

[constr_3249] Category of `HwElement` for `SwcToEcuMapping` ⌈The `HwElement` which is referenced from `SwcToEcuMapping` in the role `processingUnit` shall be of category "ProcessingUnit".

⌋*()*

[constr_3250] `PduTriggering.iSignalTriggering` condition ⌈A `PduTriggering` shall reference an `ISignalTriggering` if the `ISignalTriggering` references an `ISignal` or an `ISignalGroup` that is referenced by an `ISignalToIPduMapping` which in turn is aggregated by the `Pdu` that is referenced by that `PduTriggering`.

⌋*()*

[constr_3251] Value of `GlobalTimeDomain.domainId` in `globalTimeSubDomain` chains ⌈In a chain of `GlobalTimeDomain.globalTimeSubDomain` the value of the attribute `GlobalTimeDomain.domainId` shall be identical.

⌋*()*

[constr_3252] `ISignalTriggering.iSignalPort` reference condition ⌈An `ISignalTriggering` shall only reference an `ISignalPort` if the `CommunicationConnector` aggregating that `ISignalPort` is referenced by the `PhysicalChannel` which in turn aggregates that `ISignalTriggering`.

⌋*()*

[constr_3253] `PduTriggering.iPduPort` reference condition ⌈A `PduTriggering` shall only reference an `IPduPort` if the `CommunicationConnector` aggregating that `IPduPort` is referenced by the `PhysicalChannel` which in turn aggregates that `PduTriggering`.

⌋*()*

[constr_3254] `FrameTriggering.framePort` reference condition ⌈A `FrameTriggering` shall only reference a `FramePort` if the `CommunicationConnector` aggregating that `FramePort` is referenced by the `PhysicalChannel` which in turn aggregates that `FrameTriggering`.

⌋*()*

[constr_3255] `FrameTriggering.pduTriggering` reference condition with regard to the `PhysicalChannel` ⌈A `FrameTriggering` shall only reference a `PduTriggering` in the role `pduTriggering` if both the `FrameTriggering` and `PduTriggering` are aggregated by the same `PhysicalChannel`.

⌋*()*

[constr_3256] `PduTriggering.iSignalTriggering` reference condition with regard to the `PhysicalChannel` ⌈A `PduTriggering` shall only reference an `ISignalTriggering` in the role `iSignalTriggering` if both the `PduTriggering` and `ISignalTriggering` are aggregated by the same `PhysicalChannel`.

⌋*()*

[constr_3257] TimeSyncTechnology of servers and clients in a time synchronized network. ⌈`TimeSyncClientConfiguration.timeSyncTechnology` shall have the same value as the `TimeSyncServerConfiguration.timeSyncTechnology` that is referenced in the `TimeSyncClientConfiguration.orderedMaster` list.

⌋*()*

[constr_3258] Restriction on `ISignal.length` in case `iSignalType` is set to `array` ⌈If `ISignal.iSignalType` is set to `array` then `ISignal.length` shall be a multiple of 8.

⌋*()*

[constr_3261] `GlobalTimeDomain.pduTriggering` category ⌈The `Pdu` that is referenced by the `PduTriggering` that in turn is referenced by `GlobalTimeDomain` in the role `pduTriggering` shall be a `GeneralPurposePdu` of category GLOBAL_TIME.

⌋*()*

[constr_3262] `ConsumedEventGroup.eventGroupIdentifier` is mandatory ⌈The `ConsumedEventGroup.eventGroupIdentifier` is mandatory.

⌋*()*

[constr_3263] Restriction of usage of `SwcToEcuMapping` in a `System` ⌈For all `SwcToEcuMapping`s in a `System` the following restriction applies: No two `SwcToEcuMapping`s shall have the exact same reference to

- `SwComponentPrototype`

- `EcuInstance`

- `processingUnit`

- `controlledHwElement`

⌋*()*

[constr_3264] Server side `ClientServerToSignalMapping`s in case of a n:1 inter-ECU client-server communication ⌈If within the `System` with `category` SYSTEM_DESCRIPTION or SYSTEM_EXTRACT the `ClientServerToSignalMapping`s for inter-ECU n:1 client-server communication are placed on the provider (server) side, then each of these `ClientServerToSignalMapping`s shall (in the hierarchy of `SwComponentPrototype`s) refer to a "unique communication path" w.r.t. the `EcuInstance`s the client `SwComponentPrototype`s are mapped to.

⌋*()*

[constr_3265] `TransformationTechnology.hasInternalState` setting for an E2E transformer ⌈The value of `hasInternalState` shall be set to true for a `TransformationTechnology` with `transformerClass` set to `safety`.

⌋*()*

[constr_3266] `TransformationTechnology.hasInternalState` setting for a SOME/IP Transformer ⌈The value of `hasInternalState` shall be set to true for a SOME/IP Transformer if `SOMEIPTransformationISignalProps.sessionHandlingSR` for the `ISignal` is set to active.

⌋*()*

[constr_3267] `PduTriggering`s in Service Discovery `StaticSocketConnection`s ⌈SD `StaticSocketConnection`s defined in [TPS_SYST_02119] shall only refer to `PduTriggering`s which point to `GeneralPurposePdu`s of category SD.

⌋*()*

[constr_3268] Service Discovery `StaticSocketConnection` aggregation by a `SocketAddress` ⌈Each SD `StaticSocketConnection` defined in [TPS_SYST_02119] shall be aggregated by a `SocketAddress` that in turn aggregates an `ApplicationEndpoint` that defines a Udp Port.

⌋*()*

[constr_3269] Service Discovery `StaticSocketConnection` `remoteAddress` reference to a `TpPort` ⌈Each SD `StaticSocketConnection` defined in [TPS_SYST_02119] shall refer with the `remoteAddress` reference to an `ApplicationEndpoint` with Udp Port `portNumber` set to 0. This means that the port number is dynamically assigned at runtime.

⌋*()*

[constr_3270] Service Discovery `SocketConnection` `remoteAddress` reference to an IP Address ⌈Each SD `StaticSocketConnection` defined in [TPS_SYST_02119] shall refer with the `remoteAddress` reference to an `ApplicationEndpoint` that points to a `NetworkEndpoint` that defines an IP Address ANY (IPv4 or IPv6).

⌋*()*

[constr_3272] `SoConIPduIdentifier.headerId` **setting for SD** `StaticSocketConnection`**s** ⌈The `SoConIPduIdentifier.headerId` of SD `StaticSocketConnection`s defined in [TPS_SYST_02119] shall always be set to 0xFFFF8100 for SD messages.

⌋*()*

[constr_3273] Service Discovery multicast `StaticSocketConnection`**'s aggregation by an** `ApplicationEndpoint` ⌈The SD `StaticSocketConnection` for multicast defined in [TPS_SYST_02119] shall be aggregated by an `ApplicationEndpoint` that points to a `NetworkEndpoint` that defines an IP Multicast Address.

⌋*()*

[constr_3274] Service Discovery unicast `StaticSocketConnection`**'s aggregation by an** `ApplicationEndpoint` ⌈The SD `StaticSocketConnection` for unicast defined in [TPS_SYST_02119] shall be aggregated by an `ApplicationEndpoint` that points to a `NetworkEndpoint` that defines an IP Unicast Address.

⌋*()*

[constr_3276] Prohibition of usage of `allowedIPv6ExtHeaders` **in IPv4** `SocketAddress` ⌈IPv4 `SocketAddress` shall not define `allowedIPv6ExtHeaders`. An IPv4 `SocketAddress` aggregates an `ApplicationEndpoint` that refers to a `NetworkEndpoint` that has an `Ipv4Configuration` as `networkEndpointAddress`.

⌋*()*

[constr_3277] Restriction of usage of `IPv6ExtHeaderFilterList`**s in IPv6** `SocketAddress` ⌈All `SocketAddress`es related to the same IPv6 `NetworkEndpoint` shall all reference either no or exactly the same `IPv6ExtHeaderFilterList` with the `allowedIPv6ExtHeaders` attribute.

⌋*()*

[constr_3278] Usage of `SOMEIPTransformationProps.sizeOfArrayLengthField` ⌈The attribute `sizeOfArrayLengthField` of `SOMEIPTransformationProps` shall only be defined if the `DataPrototypeTransformationProps` is defined for a static size array according to [TPS_SYST_02121].

⌋*()*

[constr_3279] Usage of `SOMEIPTransformationProps.sizeOfStructLengthField` ⌈The attribute `sizeOfStructLengthField` of `SOMEIPTransformationProps` shall only be defined if the `DataPrototypeTransformationProps` is defined for a structure according to [TPS_SYST_02121].

⌋*()*

[constr_3280] Usage of `SOMEIPTransformationProps.sizeOfUnionLengthField` ⌈The attribute `sizeOfUnionLengthField` of `SOMEIPTransformation-`

`Props` shall only be defined if the `DataPrototypeTransformationProps` is defined for a union according to [TPS_SYST_02121].

⌋*()*

[constr_3281] Usage of `SOMEIPTransformationProps.alignment` ⌈The attribute `alignment` of `SOMEIPTransformationProps` shall only be defined if the `DataPrototypeTransformationProps` is defined for a variable data length data element according to [TPS_SYST_02121].

⌋*()*

[constr_3282] SOME/IP Transformation settings for arrays in the context of an `ISignal` ⌈In the context of an `ISignal` the usage of `DataPrototypeTransformationProps.transformationProps.sizeOfArrayLengthField` is only allowed if the `SOMEIPTransformationISignalProps.sizeOfArrayLengthFields` is not defined.

⌋*()*

[constr_3283] SOME/IP Transformation settings for structures in the context of an `ISignal` ⌈In the context of an `ISignal` the usage of `DataPrototypeTransformationProps.transformationProps.sizeOfStructLengthField` is only allowed if the `SOMEIPTransformationISignalProps.sizeOfStructLengthFields` is not defined.

⌋*()*

[constr_3284] SOME/IP Transformation settings for unions in the context of an `ISignal` ⌈In the context of an `ISignal` the usage of `DataPrototypeTransformationProps.transformationProps.sizeOfUnionLengthField` is only allowed if the `SOMEIPTransformationISignalProps.sizeOfUnionLengthFields` is not defined.

⌋*()*

[constr_3285] Alignment of variable data length data elements in the context of an `ISignal` ⌈The definition of `DataPrototypeTransformationProps.transformationProps.alignment` is only allowed if the `SOMEIPTransformationDescription.alignment` is not defined.

⌋*()*

[constr_3297] Prohibition of usage of `allowedTcpOptions` in Udp `SocketAddress` ⌈Udp `SocketAddress` shall not define `allowedTcpOptions`. A Udp `SocketAddress` aggregates an `ApplicationEndpoint` that has a `UdpTp` defined as `tpConfiguration`.

⌋*()*

[constr_3298] **Ipv6Configuration.ipv6Address range in case of enableAnycast** ⌈If Ipv6Configuration.enableAnycast is set to true then the Ipv6Configuration.ipv6Address needs to be in the unicast addressing range.

⌋*()*

[constr_3299] **SocketAddress.pathMtuDiscoveryEnabled setting dependency** ⌈SocketAddress.pathMtuDiscoveryEnabled shall only be set to TRUE if EthernetCommunicationConnector.pathMtuEnabled == TRUE.

⌋*()*

[constr_3311] Usage of SocketAddress.flowLabel ⌈SocketAddress.flowLabel shall only be used if the aggregated ApplicationEndpoint refers to a NetworkEndpoint with an Ipv6Configuration.

⌋*()*

[constr_3312] Consistency of vlanPriority and EthernetCommunicationConnector ⌈A GlobalTimeEthMaster refers to an EthernetCommunicationConnector in the role communicationConnector. If that EthernetCommunicationConnector is referenced by an EthernetPhysicalChannel in the role commConnector and the EthernetPhysicalChannel has a vLan tag defined via the VlanConfig then the GlobalTimeDomain of the GlobalTimeEthMaster shall aggregate EthGlobalTimeDomainProps in the role globalTimeDomainProperty and the attribute EthGlobalTimeDomainProps.vlanPriority shall exist.

⌋*()*

[constr_3313] E2E transformer configuration ⌈For each TransformationDescription variant that is a EndToEndTransformationDescription

- attribute protocol of TransformationTechnology shall be set to E2E
- attribute version of TransformationTechnology shall be set to 1.0.0
- attribute transformerClass of TransformationTechnology shall be set to safety

⌋*()*

[constr_3316] Allowed values for EndToEndTransformationDescription.maxDeltaCounter in PROFILE_07, PROFILE_08, PROFILE_07m and PROFILE_08m ⌈If the EndToEndTransformationDescription.profileName attribute has a value of PROFILE_07, PROFILE_08, PROFILE_07m, or PROFILE_08m the value of maxDeltaCounter attribute shall be in the range 1-4'294'967'295.

⌋*()*

[constr_3317] Assuring the same data interpretation on the sender and receiver sides in case of serialization based on the ImplementationDataTypes ⌈In order to assure the same interpretation of the serialized data by the SOME/IP transformers

on the sender and receiver sides in case of serialization based on either a primitive or a composite `ImplementationDataType`, the same `SwBaseType` shall be defined

- for this primitive `DataPrototype` or
- for each primitive `DataPrototype` of the leaf elements of the composite `DataPrototype` starting from the first element until and including the last element that is requested by the receiver,

by the `ImplementationDataType`s that either types the corresponding `PortPrototype`s on the top level Software Composition of the communicating `EcuInstance`s, or it is mapped to the `ApplicationDataType` that types it.

⌋*()*

[constr_3318] Allowed use of `ISignal.networkRepresentationProps` ⌈If a reference from `ISignal` to `DataTransformation` in the role `dataTransformation` exists, this `ISignal` SHALL NOT aggregate `SwDataDefProps` in the role `networkRepresentationProps`.

⌋*()*

[constr_3319] Existence of `DataPrototypeTransformationProps.networkRepresentationProps` ⌈`ISignal.transformationISignalProps.dataPrototypeTransformationProps.networkRepresentationProps` shall either

- not exist at all or
- shall be defined for all leaf elements of the root `DataPrototype` transmitted in the `ISignal`

⌋*()*

[constr_3322] Consistent setting of `SoConIPduIdentifier.pduCollectionSemantics` in the context of one `SocketAddress` ⌈The value of the attribute `SoConIPduIdentifier.pduCollectionSemantics` shall be identical for all referenced `SoConIPduIdentifier`s within the context of a given `SocketAddress`.

⌋*()*

[constr_3323] Relation between `NmCluster.nmPncParticipation` and `PncMapping.pncGroup` ⌈If a `PncMapping` references an `ISignalIPduGroup` in role `pncGroup` which in turn

- contains (either directly or via one of its subordinate `ISignalIPduGroup`s referenced in role `containedISignalIPduGroup`) `ISignalIPdu`s that are referenced by a `PduTriggering` in role `iPdu` which in turn
- is composed by a `PhysicalChannel` in role `pduTriggering` which in turn
- is composed by `CommunicationCluster` in role `physicalChannel` which in turn

- is referenced by an `NmCluster` in role `communicationCluster`,

then this `NmCluster` shall have its `nmPncParticipation` attribute set to TRUE unless the `PhysicalChannel` is referenced in the role `managedPhysicalChannel`.

⌋*()*

[constr_3324] **Category of `SecureCommunicationFreshnessProps` and `SecureCommunicationAuthenticationProps`** ⌈`SecureCommunicationFreshnessProps` that is referenced by a `SecuredIPdu` in the role `freshnessProps` shall have the same `category` value as the `SecureCommunicationAuthenticationProps` that is referenced by the same `SecuredIPdu` in the role `authenticationProps`.

⌋*()*

[constr_3325] **`SecureCommunicationFreshnessProps`, `SecureCommunicationAuthenticationProps` and `CryptoServicePrimitive` attribute value settings for standardized AUTOSAR security profiles** ⌈

Attributes	PROFILE_01	PROFILE_02	PROFILE_03
`algorithmFamily`	CRYPTO_ALGOFAM_AES	CRYPTO_ALGOFAM_AES	CRYPTO_ALGOFAM_AES
`algorithmMode`	CRYPTO_ALGOMODE_CMAC	CRYPTO_ALGOMODE_CMAC	CRYPTO_ALGOMODE_CMAC
`length`	128 bits	128 bits	128 bits
`authInfoTxLength`	24 bits	24 bits	28 bits
`freshnessValueLength`	Not specified	0 bits	64 bits
`freshnessValueTxLength`	8 bits	0 bits	4 bits

⌋*()*

[constr_3326] **Allowed values for `EndToEndTransformationDescription.dataIdMode` in PROFILE_11** ⌈If the `EndToEndTransformationDescription.profileName` attribute has a value of PROFILE_11 then the value of the `EndToEndTransformationDescription.dataIdMode` attribute shall be set to `all16Bit` or `lower12Bit`.

⌋*()*

[constr_3327] **Effect of `EndToEndTransformationDescription.upperHeaderBitsToShift` value in PROFILE_22** ⌈If the `EndToEndTransformationDescription.profileName` attribute has a value of PROFILE_22 and the serializing transformer is different than the ComBasedTransformer, then `EndToEndTransformationDescription.offset` shall be set to the same value of `upperHeaderBitsToShift`.

⌋*()*

[constr_3328] **`SomeipTpConnection.transportPdu` reference restriction** ⌈A `PduTriggering` that is referenced by a `SomeipTpConnection` in the role `trans-`

`portPdu` shall reference a `GeneralPurposeIPdu` with category SOMEIP_SEG-MENTED_IPDU in the role `iPdu`.

⌋*()*

[constr_3329] `SomeipTpConnection.tpSdu` reference restriction ⌈A `PduTriggering` that is referenced by a `SomeipTpConnection` in the role `tpSdu` shall reference an `IPdu` in the role `iPdu`.

⌋*()*

[constr_3330] Same `transportPdu` shall not be used in different `SomeipTpConnection`s ⌈A `PduTriggering` that is referencing a `GeneralPurposeIPdu` with category SOMEIP_SEGMENTED_IPDU in the role `iPdu` shall be referenced at most once by a `SomeipTpConnection` in the role `transportPdu`.

⌋*()*

[constr_3331] Standardized values for the attribute `category` of meta-class `EthernetCommunicationConnector` ⌈The following values of the attribute `category` of meta-class `EthernetCommunicationConnector` are reserved by the AUTOSAR standard:

- WIRED: This represents the usage of the `EthernetCommunicationConnector` in case of a wired ethernet connection
- WIRELESS: This represents the usage of the `EthernetCommunicationConnector` in case of a wireless ethernet connection
- CAN_XL: This represents the tunneling of Ethernet frames handled by the `EthernetCommunicationConnector` through CAN XL.

⌋*()*

[constr_3332] Standardized values for the attribute `category` of meta-class `EthernetCommunicationController` ⌈The following values of the attribute `category` of meta-class `EthernetCommunicationController` are reserved by the AUTOSAR standard:

- WIRED: This represents the usage of the `EthernetCommunicationController` in case of a wired ethernet connection
- WIRELESS: This represents the usage of the `EthernetCommunicationController` in case of a wireless ethernet connection
- CAN_XL: This represents the tunneling of Ethernet frames handled by the `EthernetCommunicationController` through CAN XL.

⌋*()*

[constr_3333] Standardized values for the attribute `category` of meta-class `EthernetPhysicalChannel` ⌈The following values of the attribute `category` of meta-class `EthernetPhysicalChannel` are reserved by the AUTOSAR standard:

- WIRED: This represents the usage of the `EthernetPhysicalChannel` in case of a wired ethernet connection

- WIRELESS: This represents the usage of the `EthernetPhysicalChannel` in case of a wireless ethernet connection

⌋*()*

[constr_3334] Allowed references between `EthernetPhysicalChannel` and `EthernetCommunicationConnector` ⌈An `EthernetPhysicalChannel` is only allowed to reference `EthernetCommunicationConnector`s in the role `commConnector` that have the same `category` value as the referencing `EthernetPhysicalChannel`.

⌋*()*

[constr_3335] Allowed references between `EthernetCommunicationConnector` and `EthernetCommunicationController` ⌈An `EthernetCommunicationConnector` is only allowed to reference an `EthernetCommunicationController` in the role `commController` that has the same `category` value as the referencing `EthernetCommunicationConnector`.

⌋*()*

[constr_3336] `EthernetPhysicalChannel.soAdConfig` in case of WIRELESS `EthernetPhysicalChannel` ⌈If `EthernetPhysicalChannel` has the `category` *WIRELESS* then the `EthernetPhysicalChannel` shall not aggregate the `SoAdConfig`.

⌋*()*

[constr_3337] `IPduPort.useAuthDataFreshness` is configurable on the receiver side ⌈The `IPduPort.useAuthDataFreshness` attribute shall only be used in `IPduPort`s with the `communicationDirection` = in.

⌋*()*

[constr_3338] `IPduPort.useAuthDataFreshness` validness ⌈The `IPduPort.useAuthDataFreshness` information is only valid for `SecuredIPdu`s.

⌋*()*

[constr_3339] Relation between `authDataFreshnessStartPosition`, `authDataFreshnessLength` and `useAuthDataFreshness` ⌈If `authDataFreshnessStartPosition` and `authDataFreshnessLength` are set to a value for a `SecuredIPdu` then the `useAuthDataFreshness` shall be set as well to a value on all `IPduPort`s with `communicationDirection` = in that are referenced by a `PduTriggering` of the `SecuredIPdu`.

⌋*()*

[constr_3364] `headerLength` **shall be a multiple of 8** ⌈The header length in bits specified by `headerLength` shall be a multiple of 8.

⌋*()*

[constr_3365] `EthernetPhysicalChannel`**s with different** `category` **values are not allowed within an** `EthernetCluster` ⌈A mix of `EthernetPhysicalChannel`s with different `category` values within an `EthernetCluster` is currently not supported by AUTOSAR.

⌋*()*

[constr_3373] Limitation on the number of `PhysicalChannel`**s that are referencing a** `CommunicationConnector` ⌈A `CommunicationConnector` shall only be referenced by at most one `PhysicalChannel`.

⌋*()*

[constr_3378] **Maximal one** `AliasNameAssignment` **allowed per** `FlatInstanceDescriptor` ⌈In a given instance of `AliasNameSet` in the bound system there shall be at most one `aliasName` per `FlatInstanceDescriptor`.

⌋*()*

[constr_3379] Multiple `SocketAddress` **entries with the same IP Address, Protocol and Port in the context of a given** `EcuInstance` ⌈If there are two or more `SocketAddress` entities within the scope of one `SoAdConfig` in the scope of one `EcuInstance` that have the same static (fixed at configuration time) IP Address, Protocol and Port in the aggregated `ApplicationEndpoint` and `NetworkEndpoint`, (e.g., 192.168.1.1, Tcp and 10000, respectively) then only one of these `SocketAddress` elements shall be referenced by `ProvidedServiceInstance`s/`ConsumedServiceInstance`s in the role `localUnicastAddress`.

⌋*()*

[constr_3383] **Standardized values for the attribute** `category` **of meta-class** `GeneralPurposeConnection` ⌈The following values of the attribute `category` of meta-class `GeneralPurposeConnection` are reserved by the AUTOSAR standard:

- XcpChannel

⌋*()*

[constr_3384] `PduTriggering`**s referenced by** `GeneralPurposeConnection` **shall be defined on the same** `PhysicalChannel` ⌈The `PduTriggering`s that are referenced by the `GeneralPurposeConnection` in the role `pduTriggering` shall be defined on the same `PhysicalChannel`.

⌋*()*

[constr_3385] XcpChannel is allowed to reference exactly two `PduTriggering`**s** ⌈In case that the `category` of meta-class `GeneralPurposeConnection` is set to the

value *XcpChannel* the `GeneralPurposeConnection` is allowed to reference exactly two `PduTriggering`s in the role `pduTriggering`.

⌋*()*

[constr_3386] XcpChannel is only allowed to reference `PduTriggering`s of `GeneralPurposeIPdu`s with category XCP ⌈In case that the `category` of metaclass `GeneralPurposeConnection` is set to the value *XcpChannel* the `GeneralPurposeConnection` is allowed to reference `PduTriggering`s of `GeneralPurposeIPdu`s with category XCP.

⌋*()*

[constr_3399] Existence of `securedAreaOffset` and `securedAreaLength` ⌈If the `securedAreaOffset` is defined then the `securedAreaLength` shall be defined as well and vice versa.

⌋*()*

[constr_3402] Mandatory `offset` if `noHeader` is used ⌈For each `IPdu` which is assigned to a `ContainerIPdu` in the role `ContainerIPdu.containedPduTriggering` or `ContainerIPdu.containedIPduTriggeringProps.containedPduTriggering` with `ContainerIPdu.headerType` = `noHeader` the `ContainedIPduProps.offset` shall be defined.

⌋*()*

[constr_3403] Usage of `ContainerIPdu.rxAcceptContainedIPdu` if `noHeader` is used ⌈If the `ContainerIPdu.headerType` is set to `noHeader` then the `ContainerIPdu.rxAcceptContainedIPdu` attribute value shall be set to `acceptConfigured`.

⌋*()*

[constr_3404] Usage of `ContainedIPduProps.updateIndicationBitPosition` ⌈`ContainedIPduProps.updateIndicationBitPosition` is only allowed to be set to a value if the `headerType` of the `ContainerIPdu` that contains the `IPdu` with `ContainerIPdu.containedPduTriggering` or `ContainerIPdu.containedIPduTriggeringProps.containedPduTriggering` is set to `noHeader`.

⌋*()*

[constr_3405] Dynamic Length `IPdu` inside of a static configured `ContainerIPdu` ⌈Only the last contained `IPdu` (according to the `ContainedIPduProps.offset`) of a `ContainerIPdu` with static container layout (i.e., a `ContainerIPdu` with `headerType` set to `noHeader`) is allowed to be a dynamic length `IPdu` (i.e, a contained `IPdu` that at runtime may exhibit a length different from the one statically configured via `Pdu.length` of the respective `Pdu`). All other contained `IPdu`s of a `ContainerIPdu` with static container layout have to be static length `IPdu`s.

⌋*()*

[constr_3406] All signals before `authDataFreshnessStartPosition` shall have a static length ⌈In case that

- an `ISignalIPdu` is referenced by the `SecuredIPdu` with the `payload` reference via the `PduTriggering` and

- the `authDataFreshnessStartPosition` and `authDataFreshnessLength` define the area in the `ISignalIPdu` that is taken to verify and generate the Freshness then

all `ISignal`s that are mapped into the `ISignalIPdu` in front of the configured `authDataFreshnessStartPosition` shall have a static length.

⌋*()*

[constr_3407] Freshness Value in Authentic `IPdu` is not allowed to be used in case of `ContainerIPdu` with a dynamic layout ⌈If a `ContainerIPdu` that is referenced by the `SecuredIPdu` with the `payload` reference via the `PduTriggering` contains a dynamic layout (i.e. `ContainerIPdu.headerType` is set to `longHeader` or `shortHeader`) and multiple contained `IPdu`s then each `IPduPort` that is referenced by the `PduTriggering` of the `SecuredIPdu` shall have the attribute `useAuthDataFreshness` set to false.

⌋*()*

[constr_3435] Applicability of `CouplingPort.macMulticastAddress` ⌈The reference `CouplingPort.macMulticastAddress` is only applicable if the `CouplingPort` is aggregated by a `CouplingElement` with `couplingType` = `switch`.

⌋*()*

[constr_3436] Value range of `minimumTxContainerQueueSize` and `minimumRxContainerQueueSize` ⌈If defined, the value of `minimumTxContainerQueueSize` and `minimumRxContainerQueueSize` shall be in the range of 0..255.

⌋*()*

[constr_3437] `invalidValue` defined in the context of `ISignal` ⌈The definition of `SwDataDefProps.invalidValue` aggregated by an `ISignal` in the role `networkRepresentationProps` shall only be a `NumericalValueSpecification`, `TextValueSpecification` or `ArrayValueSpecification` that aggregates elements of type `NumericalValueSpecification` or `TextValueSpecification`.

⌋*()*

[constr_3438] `timeoutSubstitutionValue` defined in the context of `ISignal` ⌈The definition of an `timeoutSubstitutionValue` in the context of an `ISignal` shall only be a `NumericalValueSpecification`, `TextValueSpecification` or `ArrayValueSpecification` that aggregates elements of type `NumericalValueSpecification` or `TextValueSpecification`.

⌋*()*

[constr_3448] Restriction for usage of `Pdu.hasDynamicLength` ⌈The `Pdu.hasDynamicLength` attribute is only relevant for `UserDefinedPdu`s, `UserDefinedIPdu`s, `J1939DcmIPdu`s.

⌋*()*

[constr_3454] Unique `headerIdLongHeader` for `acceptConfigured` ⌈For a `ContainerIPdu` with `ContainerIPdu.rxAcceptContainedIPdu` = `RxAcceptContainedIPduEnum.acceptConfigured` and `ContainerIPdu.headerType` = `longHeader` the following shall apply: All referenced `IPdu`s (via `ContainerIPdu.containedPduTriggering` or `ContainerIPdu.containedIPduTriggeringProps.containedPduTriggering`) shall have a unique `ContainedIPduProps.headerIdLongHeader` within the scope of this `ContainerIPdu`.

⌋*()*

[constr_3455] Unique `headerIdShortHeader` for `acceptConfigured` ⌈For a `ContainerIPdu` with `ContainerIPdu.rxAcceptContainedIPdu` = `RxAcceptContainedIPduEnum.acceptConfigured` and `ContainerIPdu.headerType` = `shortHeader` the following shall apply: All referenced `IPdu`s (via `ContainerIPdu.containedPduTriggering` or `ContainerIPdu.containedIPduTriggeringProps.containedPduTriggering`) shall have a unique `ContainedIPduProps.headerIdShortHeader` within the scope of this `ContainerIPdu`.

⌋*()*

[constr_3456] Existence of `ProvidedServiceInstance.loadBalancingPriority` and `ProvidedServiceInstance.loadBalancingWeight` ⌈The attributes `ProvidedServiceInstance.loadBalancingPriority` and `ProvidedServiceInstance.loadBalancingWeight` shall either not exist or be defined both.

⌋*()*

[constr_3457] Uniqueness of `ConsumedEventGroup.eventGroupIdentifier` in the scope of a `ConsumedServiceInstance` ⌈Each `ConsumedEventGroup` that is aggregated by a `ConsumedServiceInstance` shall have a unique `eventGroupIdentifier` value in the scope of the aggregating `ConsumedServiceInstance`.

⌋*()*

[constr_3458] `FlatInstanceDescriptor.rtePluginProps` shall only reference a `EcucContainerValue` representing a RteRipsPlugin ⌈`FlatInstanceDescriptor.rtePluginProps` shall only reference an `EcucContainerValue` which defines the identity of the `RTE Implementation Plug-In`. This requires that the according `EcucContainerValue`'s `definition` references a `EcucContainerDef` having a `destinationUri` set to `/AUTOSAR/EcucDestinationUriDefSets/RteRipsUriDefSet/RteRipsPlugin`

⌋*()*

[constr_3460] Full definition of `transferProperty` for group signal ⌈If at least one of the `ISignal`s belonging to an `ISignalGroup` has a `transferProperty` de-

fined (via their respective `ISignalToIPduMapping`) then
all other `ISignal`s belonging to the same `ISignalGroup` shall have a `transferProperty` defined as well.

⌋*()*

[constr_3461] TransferProperty for group signals if `ISignalGroup` has `transferProperty`=`pending` ⌈If the `ISignalToIPduMapping` refers to an `ISignalGroup` in the role `iSignalGroup` and
the `transferProperty` is set to `pending` then
the group signals of this `ISignalGroup` shall either

- have no `transferProperty` defined (via their respective `ISignalToIPduMapping`) or

- every `ISignal` belonging to the `ISignalGroup` shall have the `transferProperty`=`pending` defined.

⌋*()*

[constr_3464] Allowed Pdu type on `BusMirrorChannelMapping.targetChannel` ⌈Each `PduTriggering` that is referenced by `BusMirrorChannelMapping` in the role `targetPduTriggering` is only allowed to reference a `GeneralPurposeIPdu` of category BUS_MIRRORING.

⌋*()*

[constr_3465] Identical `BusMirrorChannel.busMirrorNetworkId` for `BusMirrorChannel`s referencing the same `PhysicalChannel` ⌈The attribute `BusMirrorChannel.busMirrorNetworkId` shall be identical in all `BusMirrorChannel`s that are referencing the same `PhysicalChannel` in the scope of the `System`.

⌋*()*

[constr_3466] Unique `BusMirrorChannel.busMirrorNetworkId`s for each specialization of `PhysicalChannel` ⌈The attribute `BusMirrorChannel.busMirrorNetworkId` associated with `PhysicalChannel`s that have the same specialization (e.g. all `CanPhysicalChannel`s) shall have unique `BusMirrorChannel.busMirrorNetworkId`s within the scope of the `System`).

⌋*()*

[constr_3467] `CanPhysicalChannel` as destination channel of `BusMirrorChannelMappingCan` ⌈The `BusMirrorChannel` that is aggregated by `BusMirrorChannelMappingCan` shall only reference a `CanPhysicalChannel` in the role `targetChannel`.

⌋*()*

[constr_3468] `BusMirrorChannelMappingCan.targetPduTriggering` restriction ⌈`BusMirrorChannelMappingCan` is allowed to reference only one single `PduTriggering` in the role `targetPduTriggering`.

⌋*()*

[constr_3469] `CanFrameTriggering.txMask` setting for the destination frame ⌈The `CanFrameTriggering` of a `Frame` that contains a `Pdu` of which the `PduTriggering` is referenced by `BusMirrorChannelMappingCan` in the role `targetPduTriggering` shall set the `txMask` to 0.

⌋*()*

[constr_3470] PaddingValue used to transmit the Pdu on a Can-Fd destination bus ⌈In case that the `BusMirrorChannelMappingCan` references a `PduTriggering` in the role `targetPduTriggering` and

- the `CanFrameTriggering` of the `Frame` that contains this `targetPduTriggering` has the `canFrameTxBehavior` set to `canFd` and

- the `CanFrameTriggering` has a reference to an "out" `FramePort` (i.e. the Frame is transmitted by an Ecu on a Can-Fd destination bus) and

- the `CommunicationController` of the transmitting `EcuInstance` that is referenced via the `CommunicationConnector` by the `PhysicalChannel` on which the `targetPduTriggering` is located then the `CanControllerFdConfiguration.paddingValue` or `CanControllerFdConfigurationRequirements.paddingValue` shall have the value 0.

⌋*()*

[constr_3471] `FlexrayPhysicalChannel` as destination channel of `BusMirrorChannelMappingFlexray` ⌈The `BusMirrorChannel` that is aggregated by `BusMirrorChannelMappingFlexray` shall only reference a `FlexrayPhysicalChannel` in the role `targetChannel`.

⌋*()*

[constr_3472] Number of `BusMirrorChannel`s derived for one `FlexrayCluster` ⌈For each `FlexrayCluster`, only one `BusMirrorChannel` shall be derived. I.e. if both channels A and B are derived, only one of the two `FlexrayPhysicalChannel`s of one `FlexrayCluster` shall be referenced by a `BusMirrorChannel` in the `System`.

⌋*()*

[constr_3473] `BusMirrorChannelMappingFlexray.targetPduTriggering` restriction ⌈The `FlexrayFrameTriggering` of a `Frame` that contains a `Pdu` of which the `PduTriggering` is referenced by `BusMirrorChannelMappingFlexray` in the

role `targetPduTriggering` shall have the `allowDynamicLSduLength` attribute set to true.

⌋*()*

[constr_3474] EthernetPhysicalChannel as destination channel of BusMirrorChannelMappingIp ⌈The `BusMirrorChannel` that is aggregated by `BusMirrorChannelMappingIp` shall only reference an `EthernetPhysicalChannel` in the role `targetChannel`.

⌋*()*

[constr_3475] BusMirrorChannelMappingIp.targetPduTriggering restriction ⌈`BusMirrorChannelMappingIp` is allowed to reference only one single `PduTriggering` in the role `targetPduTriggering`.

⌋*()*

[constr_3476] UserDefinedPhysicalChannel as destination channel of BusMirrorChannelMappingUserDefined ⌈The `BusMirrorChannel` that is aggregated by `BusMirrorChannelMappingUserDefined` shall only reference a `UserDefinedPhysicalChannel` in the role `targetChannel`.

⌋*()*

[constr_3477] BusMirrorChannelMappingUserDefined.targetPduTriggering restriction ⌈`BusMirrorChannelMappingUserDefined` is allowed to reference only one single `PduTriggering` in the role `targetPduTriggering`.

⌋*()*

[constr_3479] PhysicalChannel is not allowed to be a managedPhysicalChannel and a managing PhysicalChannel ⌈If a `PhysicalChannel` is referenced in role `managedPhysicalChannel`, then it shall not be the source of another `managedPhysicalChannel` relation.

⌋*()*

[constr_3480] PhysicalChannel shall be referenced in the role managedPhysicalChannel only once ⌈A `PhysicalChannel` shall be referenced in the role `managedPhysicalChannel` only up to once.

⌋*()*

[constr_3481] UdpNmCluster is not allowed to reference a managedPhysicalChannel in the role vlan ⌈If an `EthernetPhysicalChannel` is target of a `managedPhysicalChannel` reference, then no `UdpNmCluster` shall reference this `managedPhysicalChannel` in the role `vlan`.

⌋*()*

[constr_3482] NmCluster is not allowed to reference a CommunicationCluster that aggregates a managedPhysicalChannel ⌈If a `PhysicalChannel`, except

`EthernetPhysicalChannel`, is target of a `managedPhysicalChannel`, then the aggregating `CommunicationCluster` shall not be referenced by any `NmCluster` in the role `communicationCluster`.

⌋*()*

[constr_3484] `PncMapping` that refers a `managedPhysicalChannel` shall also refer the managing `PhysicalChannel` ⌈If a `PncMapping` refers to a `PhysicalChannel` (either directly in the role `physicalChannel` or indirectly by referencing an `ISignalIPduGroup` in the role `pncGroup`) and this `PhysicalChannel` is referenced in the role `managedPhysicalChannel`, then the according managing `PhysicalChannel` (the source of the `managedPhysicalChannel` reference) shall also be referenced by the `PncMapping` (either directly in the role `physicalChannel` or indirectly by referencing an `ISignalIPduGroup` in the role `pncGroup`).

⌋*()*

[constr_3488] Value range of `ContainedIPduProps.priority` ⌈If defined, the value of `ContainedIPduProps.priority` shall be in the range of 0..255.

⌋*()*

[constr_3489] `ContainedIPduProps.priority` is only applicable if a `ContainerIPdu` header is used ⌈`ContainedIPduProps.priority` is only applicable if the `headerType` of the `ContainerIPdu` is set to `shortHeader` or `longHeader`.

⌋*()*

[constr_3490] `ContainedIPduProps.priority` is only applicable if `collectionSemantics` is set to `lastIsBest` ⌈`ContainedIPduProps.priority` is only applicable if `ContainedIPduProps.collectionSemantics` is set to `lastIsBest`.

⌋*()*

[constr_3501] Role of `SystemSignal` in 1:n communication ⌈In case of 1:n communication the `VariableDataPrototype` in the `PPortPrototype` of the `SwComponentPrototype` shall be mapped to only one `SystemSignal`.

⌋*()*

[constr_3506] Mapping of composite data type to `SystemSignal`s in `SystemSignalGroup` ⌈Either all or a subset of elements of a composite data type shall be mapped to `SystemSignal`s which shall be members of one `SystemSignalGroup` if no data transformation (except COM Based Transformer) is used.

There are two exceptions to this rule:

- it is allowed to map an array `VariableDataPrototype` consisting of `UINT8` elements to exactly one `SystemSignal` in the context of one `SenderReceiverToSignalMapping` (see [TPS_SYST_01037]).

- in case the COM Based Transformer [11] is used it is the integral part of the approach to have a fixed mapping of the individual elements of composite data types to `SystemSignal`s in a `SystemSignalGroup` ([TPS_SYST_02058]).

⌋*()*

[constr_3508] Value of `nmReadySleepTime` ⌈The `nmReadySleepTime` value shall be a multiple of `cycle` * `nmRepetitionCycle`.

⌋*()*

[constr_3514] No two `ISignalToIPduMapping`s shall reference the identical `ISignal` ⌈No two `ISignalToIPduMapping`s shall reference the identical `ISignal` in the role `iSignal` in the scope of one System.

⌋*()*

[constr_3515] Fully filled `EthernetPriorityRegeneration` table ⌈In case the `CouplingPortDetails`.`ethernetPriorityRegeneration` is defined it shall contain exactly 8 elements of `EthernetPriorityRegeneration`, one for each value of `ingressPriority` (0-7).

⌋*()*

[constr_3516] limitation of `Frame.frameLength` for CAN L-PDUs ⌈The `Frame`.`frameLength` of CAN PDUs shall be restricted to

- 0..8 for classic CAN L-PDUs;

- 0..8, 12, 16, 20, 24, 32, 48, 64 for CAN FD L-PDUs and

- 1..2048 for CAN XL L-PDUs.

⌋*()*

[constr_3517] Consistent setting of `ContainedIPduProps.collectionSemantics` in the context of one `ContainerIPdu` ⌈The value of the attribute `ContainedIPduProps`.`collectionSemantics` shall be identical for all contained `IPdu`s within the context of a given `ContainerIPdu`.

⌋*()*

[constr_3518] Range of CanControllerFdConfiguration.paddingValue and CanControllerFdConfigurationRequirements.paddingValue ⌈The value given for `CanControllerFdConfiguration`.`paddingValue` and `CanControllerFdConfigurationRequirements`.`paddingValue` shall be in the range from 0 to 255.

⌋*()*

[constr_3519] Value of `category` of `GlobalTimeDomain` ⌈The attribute `category` of `GlobalTimeDomain` can have the following values:

- SYNCHRONIZED: this time base does not depend on the existence of another time base

- OFFSET: this time base depends on the existence of another time base. It delivers a value that represents an offset relative to the referenced (`GlobalTimeDomain.offsetTimeDomain`) synchronized time base.

⌋*()*

[constr_3520] Offset time domain shall be based on a synchronized time domain
⌈If a `GlobalTimeDomain` has a reference with the role `GlobalTimeDomain.offsetTimeDomain` the reference source shall have a `GlobalTimeDomain.domainId` in the range of 16-31 and the reference target shall have a `GlobalTimeDomain.domainId` in the range of 0-15.

⌋*()*

[constr_3521] `defaultVlan` and `vlanMembership` ⌈If a `CouplingPort` refers to an `EthernetPhysicalChannel` in the role `defaultVlan` the `CouplingPort` shall also have a `vlanMembership` defined. This `VlanMembership` shall point to the same `EthernetPhysicalChannel` in the role `vlan` as the `defaultVlan`.

⌋*()*

[constr_3522] `vlanModifier` and `vlanMembership` ⌈If a `CouplingPort` refers to an `EthernetPhysicalChannel` in the role `vlanModifier` the `CouplingPort` shall also have a `vlanMembership` defined. This `VlanMembership` shall point to the same `EthernetPhysicalChannel` in the role `vlan` as the `vlanModifier`.

⌋*()*

[constr_3523] `CouplingPort` and `PncMapping` in the scope of an `EthernetPhysicalChannel` ⌈If

- a `CouplingPort` referring to an `EthernetPhysicalChannel` – via a `VlanMembership` – references at least one `PncMapping`

- and that `PncMapping` contains PDUs – via the assignment of `PncMapping.pncGroup` – that are transported on this `EthernetPhysicalChannel`

then every `CouplingPort` referring to that `EthernetPhysicalChannel` shall reference at least one `PncMapping` as well.

⌋*()*

[constr_3524] Definition of `couplingPortRole` on `CouplingPort` for managed `CouplingElement` ⌈A managed `CouplingElement` shall have either

- at most one `CouplingPort` with `couplingPortRole` set to `hostPort` or

- at least one `CouplingPort` with `couplingPortRole` set to `upLinkPort`.

⌋*()*

[constr_3525] Connection of `CouplingPort` with `couplingPortRole` set to `upLinkPort` ⌈A `CouplingPort` with `couplingPortRole` set to `upLinkPort` shall

be connected to exactly one other `CouplingPort` with `couplingPortRole` set to `upLinkPort`.

⌋*()*

[constr_3533] **`EndToEndTransformationISignalProps.dataLength` shall be a multiple of 8** ⌈The value of `EndToEndTransformationISignalProps`.`dataLength`, `EndToEndTransformationISignalProps`.`maxDataLength`, and `EndToEndTransformationISignalProps`.`minDataLength` shall be a multiple of 8.

⌋*()*

[constr_3534] **`EthernetPhysicalChannel` shall only be referenced by one `VlanMembership`** ⌈An `EthernetPhysicalChannel` shall only be referenced by one `VlanMembership` in the role `VlanMembership`.`vlan` in the scope of one `CouplingPort`.

⌋*()*

[constr_3535] **`EthernetCommunicationController` shall aggregate at most one `CouplingPort`** ⌈An `EthernetCommunicationController` is allowed to aggregate at most one `CouplingPort`.

⌋*()*

[constr_3545] Mandatory reference to a *Pnc* in case of `anyPartialNetworkActive` or `allPartialNetworksActive` ⌈If the `SignalServiceTranslationProps`.`serviceControl` equals `anyPartialNetworkActive` or `allPartialNetworksActive`, then the reference `SignalServiceTranslationProps`.`controlPnc` shall point to at least one `PncMappingIdent`.

⌋*()*

[constr_3546] Mandatory reference to a `ConsumedEventGroup` in case of `serviceControl` ⌈For a provided translated service instance, if the `SignalServiceTranslationProps`.`serviceControl` equals `serviceDiscovery` then the reference `SignalServiceTranslationProps`.`controlConsumedEventGroup` shall point to at least one `ConsumedEventGroup`.

⌋*()*

[constr_3548] EndToEnd profile for both ends of `safeTranslation` ⌈If the `SignalServiceTranslationEventProps`.`safeTranslation` equals true then both, the signal-based payload as well as the service-oriented payload shall have an EndToEnd profile defined.

⌋*()*

[constr_3549] Secure payload for both ends in case of `secureTranslation` ⌈If the `SignalServiceTranslationEventProps`.`secureTranslation` equals true

then both, the signal-based payload as well as the service-oriented payload shall have a secure communication defined.

⌋*()*

[constr_3559]{DRAFT} **`ConsumedServiceInstance.blocklistedVersion` is restricted to the usage of `minorVersion`** ⌈The `majorVersion` attribute shall not be used in the `SomeipServiceVersion` that is aggregated by the `ConsumedServiceInstance` in the role `blocklistedVersion`.

⌋*()*

[constr_3560]{DRAFT} **`minimumMinorVersion` and `ConsumedServiceInstance.minorVersion` value** ⌈The `ConsumedServiceInstance.minorVersion` shall not have the value *ANY* if `versionDrivenFindBehavior` = `minimumMinorVersion`.

⌋*()*

[constr_3600] Setting of `EthernetCommunicationController.slaveActAsPassiveCommunicationSlave` ⌈The attribute `EthernetCommunicationController.slaveActAsPassiveCommunicationSlave` may only be set to TRUE, if the following conditions apply:

- the `EthernetCommunicationController` is not referenced by any `NmNode` in the role `controller`

- the `EthernetCommunicationController` aggregates at least one `CouplingPort`

- the `couplingPortRole` of that `CouplingPort` is set to `standardPort`

- the `physicalLayerType` of that `CouplingPort` is set to either `_100BASE_T1` or `_1000BASE_T1`

In all other cases the attribute `slaveActAsPassiveCommunicationSlave` shall be set to FALSE or shall not be defined.

⌋*()*

[constr_3601] Mandatory attributes of `EthernetWakeupSleepOnDatalineConfig` ⌈The following attributes of `EthernetWakeupSleepOnDatalineConfig` shall be defined at the time when the COM Stack is generated:

- `wakeupLocalEnabled`

- `wakeupRemoteEnabled`

⌋*()*

[constr_3602] Existence of `wakeupForwardLocalEnabled` ⌈The attribute `wakeupForwardLocalEnabled` shall be defined if `wakeupRemoteEnabled` is set to TRUE.

⌋*()*

[constr_3603] Existence of `wakeupLocalDurationTime` ⌈The attribute `wakeupLocalDurationTime` shall be defined if `wakeupForwardLocalEnabled` is set to TRUE.

⌋*()*

[constr_3604] Existence of `wakeupForwardRemoteEnabled` ⌈The attribute `wakeupForwardRemoteEnabled` shall be defined if `wakeupLocalEnabled` is set to TRUE.

⌋*()*

[constr_3605] Existence of `wakeupLocalDetectionTime` ⌈The attribute `wakeupLocalDetectionTime` shall be defined if `wakeupForwardRemoteEnabled` is set to TRUE.

⌋*()*

[constr_3606] Values of `wakeupLocalDurationTime` and `wakeupLocalDetectionTime` ⌈If defined, then the value of `wakeupLocalDurationTime` shall be greater than the value of `wakeupLocalDetectionTime`.

⌋*()*

[constr_3607] Existence of `sleepRepetitionDelayOfSleepRequest` ⌈The attribute `sleepRepetitionDelayOfSleepRequest` shall be defined if `sleepRepetitionsOfSleepRequest` is defined and has a value greater than 0.

⌋*()*

[constr_3608] Existence of `wakeupRepetitionDelayOfWakeupRequest` ⌈The attribute `wakeupRepetitionDelayOfWakeupRequest` shall only be defined if `wakeupRepetitionsOfWakeupRequest` is defined and has a value greater than 0.

⌋*()*

[constr_3609] Values of `wakeupLocalDurationTime` in the context of a `CouplingElement` ⌈All `CouplingPort`s which have the reference `wakeupSleepOnDatalineConfig` defined and

- where the `CouplingPort`s are aggregated by the same `CouplingElement` and

- where the referenced `EthernetWakeupSleepOnDatalineConfig` has the attribute `wakeupLocalDurationTime` defined

shall refer to `EthernetWakeupSleepOnDatalineConfig`s where the value of `wakeupLocalDurationTime` is identical for all referencing `CouplingPort`s.

⌋*()*

[constr_3610] Values of `wakeupLocalDetectionTime` in the context of a `CouplingElement` ⌈All `CouplingPort`s which have the reference `wakeupSleepOnDatalineConfig` defined and

- where the `CouplingPort`s are aggregated by the same `CouplingElement` and

- where the referenced `EthernetWakeupSleepOnDatalineConfig` has the attribute `wakeupLocalDetectionTime` defined

shall refer to `EthernetWakeupSleepOnDatalineConfig`s where the value of `wakeupLocalDetectionTime` is identical for all referencing `CouplingPort`s.

⌋*()*

[constr_3611] Existence of `EthernetCommunicationController.slaveQualifiedUnexpectedLinkDownTime` ⌈The attribute `slaveQualifiedUnexpectedLinkDownTime` shall be defined if `slaveActAsPassiveCommunicationSlave` is set to TRUE.

⌋*()*

[constr_3615] Existence of `EthernetCluster.couplingPortSwitchoffDelay` ⌈The attribute `EthernetCluster.couplingPortSwitchoffDelay` shall be defined if at least one `EcuInstance` connected to that `EthernetCluster` has the attribute `ethSwitchPortGroupDerivation` set to TRUE.

⌋*()*

[constr_3616] Value of `EthernetCluster.couplingPortSwitchoffDelay` ⌈If defined, the value of `EthernetCluster.couplingPortSwitchoffDelay` shall be greater than `UdpNmCluster.nmNetworkTimeout` + `UdpNmCluster.nmWaitBusSleepTime` of the respective `EthernetCluster`.

⌋*()*

[constr_3617] Existence of `EthernetCluster.couplingPortStartupActiveTime` ⌈The attribute `EthernetCluster.couplingPortStartupActiveTime` shall be defined if at least one `EcuInstance` connected to that `EthernetCluster` has the attribute `ethSwitchPortGroupDerivation` set to TRUE.

⌋*()*

[constr_3618] Value of `EthernetCluster.couplingPortStartupActiveTime` ⌈If defined, the value of `EthernetCluster.couplingPortStartupActiveTime`

shall be greater than `UdpNmCluster`.nmNetworkTimeout + `UdpNmCluster`.nmWaitBusSleepTime of the respective `EthernetCluster`.

⌋*()*

[constr_3620]`GlobalTimeDomain.networkSegmentId`only applicable to Global Time sub domains ⌈The aggregation `GlobalTimeDomain`.networkSegmentId shall only be defined if the `GlobalTimeDomain` is itself referenced in the role `GlobalTimeDomain`.globalTimeSubDomain.

⌋*()*

[constr_3621] Value range of `GlobalTimeDomain.networkSegmentId` ⌈If defined, the value of `GlobalTimeDomain`.networkSegmentId shall be in the range 0..255.

⌋*()*

[constr_3651] No `element` in case `translationTarget` is primitive ⌈If `SignalServiceTranslationEventProps`.translationTarget refers to a `VariableDataPrototype` that is typed by a primitive `AutosarDataType` then the reference `SignalServiceTranslationElementProps`.element shall not be defined.

⌋*()*

[constr_3652] Allowed sub-classes of `DataPrototypeReference` in the context of `signal/service translation` ⌈If a `DataPrototypeReference` in the role `SignalServiceTranslationElementProps`.element is used then following sub-classes are supported:

- if the reference target is typed by an `ApplicationDataType` then the `DataPrototypeInSenderReceiverInterfaceInstanceRef` shall be used and shall target an `ApplicationCompositeElementDataPrototype`.
- if the reference target is typed by an `ImplementationDataType` then the `ImplementationDataTypeElementInPortInterfaceRef` shall be used.

⌋*()*

[constr_3653] Consistent `translationTarget` and `element` in case `ApplicationDataType` is used ⌈If the `SignalServiceTranslationEventProps`.translationTarget refers to a `VariableDataPrototype` that is typed by an `ApplicationDataType` (targetDataPrototype of the `VariableDataPrototypeInSystemInstanceRef`) then every `SignalServiceTranslationElementProps`.element reference that is defined in the context of the `SignalServiceTranslationEventProps` shall have that `VariableDataPrototype` as the rootDataPrototypeInSr of the `DataPrototypeInSenderReceiverInterfaceInstanceRef`.

⌋*()*

[constr_3654] Consistent `translationTarget` and `element` in case `ImplementationDataType` is used ⌈If the `SignalServiceTranslationEventProps.translationTarget` refers to a `VariableDataPrototype` that is typed by an `ImplementationDataType` (`targetDataPrototype` of the `VariableDataPrototypeInSystemInstanceRef`) then every `SignalServiceTranslationElementProps.element` reference that is defined in the context of the `SignalServiceTranslationEventProps` shall have that `VariableDataPrototype` as the `rootDataPrototype` of the `ImplementationDataTypeElementInPortInterfaceRef`.

⌋*()*

[constr_3655] Supported filter types for primitive `SignalServiceTranslationElementProps` ⌈If the target for `SignalServiceTranslationElementProps` is defined as primitive according to [TPS_SYST_03062] then the following values for `dataFilterType` are supported:

- `always`
- `maskedNewDiffersMaskedOld`
- `maskedNewDiffersX`
- `maskedNewEqualsX`
- `never`
- `newIsOutside`
- `newIsWithin`
- `oneEveryN`.

⌋*()*

[constr_3656] Supported filter types for composite `SignalServiceTranslationElementProps` ⌈If the target for `SignalServiceTranslationElementProps` is defined as composite according to [TPS_SYST_03062] then the following values for `dataFilterType` are supported:

- `always`
- `never`
- `oneEveryN`.

⌋*()*

[constr_3668] Existence of `TlsCryptoCipherSuite.cipherSuiteShortLabel` ⌈If a `TlsCryptoCipherSuite.cipherSuiteShortLabel` is defined then:

- the attribute `TlsCryptoCipherSuite.cipherSuiteId` shall be defined as well

- the value of `TlsCryptoCipherSuite.cipherSuiteShortLabel` shall match the *Description* value corresponding to the *Value* field defined in `TlsCryptoCipherSuite.cipherSuiteId` according to `TlsCryptoCipherSuite` Parameter set defined in [13].

⌋*()*

[constr_3669] `eventMulticastSubscriptionAddress` shall refer to a multicast address ⌈The reference `ConsumedServiceInstance.eventMulticastSubscriptionAddress` shall refer to an `ApplicationEndpoint` which in turn refers to a `NetworkEndpoint` that represents a multicast address.

⌋*()*

[constr_3670] No support for parallel `localUnicastAddress` and `eventMulticastSubscriptionAddress` ⌈If a `eventMulticastSubscriptionAddress` is defined for a `ConsumedServiceInstance` then there shall not be a `localUnicastAddress` defined at the same `ConsumedServiceInstance`.

⌋*()*

[constr_3671] `remoteMulticastSubscriptionAddress` shall refer to a multicast address ⌈The reference `ProvidedServiceInstance.remoteMulticastSubscriptionAddress` shall refer to an `ApplicationEndpoint` which in turn refers to a `NetworkEndpoint` that represents a multicast address.

⌋*()*

[constr_3672] No support for methods in `multicast subscription` at the client ⌈If a `ConsumedServiceInstance` aggregates a `PduActivationRoutingGroup` in the role `methodActivationRoutingGroup`, then the `ConsumedServiceInstance` shall not define a `eventMulticastSubscriptionAddress`.

⌋*()*

[constr_3673] No support for methods in `multicast subscription` at the server static configuration ⌈If a `ProvidedServiceInstance` aggregates a `PduActivationRoutingGroup` in the role `methodActivationRoutingGroup`, then the `ProvidedServiceInstance.remoteMulticastSubscriptionAddress` shall not be defined.

⌋*()*

[constr_3685] Allowed values for each element of `pncFilterArrayMask` ⌈The value for each element of `CommunicationConnector.pncFilterArrayMask` shall be in the range between 0 and 255.

⌋*()*

[constr_3686] Allowed number of entries for `pncFilterArrayMask` ⌈The number of `CommunicationConnector.pncFilterArrayMask` elements shall be:

- `NmCluster.pncClusterVectorLength`, if defined

- `System.pncVectorLength`, otherwise.

⌋*()*

[constr_3687] Limited value range for `NmCluster.pncClusterVectorLength`
⌈The value of `NmCluster.pncClusterVectorLength` shall be equal or smaller than `System.pncVectorLength`.

⌋*()*

[constr_3695] `canControllerXlAttributes` and `canControllerXlRequirements` are mutually exclusive ⌈The existence of `canControllerXlAttributes` and `canControllerXlRequirements` is mutually exclusive.

⌋*()*

[constr_3696] Mandatory attributes of `CanControllerXlConfiguration` ⌈A `CanControllerConfiguration` configuring a CAN XL controller shall aggregate `CanControllerXlConfiguration` with the following attributes defined at the time when the COM Stack is generated:

- `errorSignalingEnabled`
- `propSeg`
- `syncJumpWidth`
- `timeSeg1`
- `timeSeg2`
- `trcvPwmModeEnabled`

⌋*()*

[constr_3697] Latest existence time of `CanControllerXlConfiguration` and `CanControllerXlConfigurationRequirements` ⌈At completion time of the System Description, either `CanControllerXlConfiguration` or `CanControllerXlConfigurationRequirements` shall exist within their aggregating class `CanControllerConfiguration` or `CanControllerConfigurationRequirements`, respectively.

⌋*()*

[constr_3698] Value of `errorSignalingEnabled` ⌈The attribute `errorSignalingEnabled` shall be set to FALSE if `trcvPwmModeEnabled` is set to TRUE.

⌋*()*

[constr_3699] Existence of `pwmL` ⌈The attribute `pwmL` shall be defined if `trcvPwmModeEnabled` is set to TRUE.

⌋*()*

[constr_3700] Existence of `pwmO` ⌈The attribute `pwmO` shall be defined if `trcvPwm-ModeEnabled` is set to TRUE.

⌋*()*

[constr_3701] Existence of `pwmS` ⌈The attribute `pwmS` shall be defined if `trcvPwm-ModeEnabled` is set to TRUE.

⌋*()*

[constr_3702] Relevant attributes of EthernetCommunicationController for CAN XL ⌈If the category of `EthernetCommunicationController` is equal to CAN_XL, then only the following attributes of this meta-class are relevant:

- `macLayerType`

- `macUnicastAddress`

⌋*()*

[constr_3703] Reference to CanControllerXlConfiguration in case of category CAN_XL ⌈If the category of `EthernetCommunicationController` is equal to CAN_XL, then the reference `canXlConfig` of `EthernetCommunicationController` shall refer to the `CanCommunicationController` aggregating the `CanControllerConfiguration` which in turn aggregates the `CanControllerXlConfiguration` that is used for tunneling of the Ethernet frames associated with the aforementioned `EthernetCommunicationController`.

⌋*()*

[constr_3704] Existence of `CanXlFrameTriggeringProps` ⌈If the class `CanXlFrameTriggeringProps` is aggregated by a `CanFrameTriggering`, then the `CanCommunicationController` – which is referenced through `commController` by a `CanCommunicationConnector` which in turn is referenced through `commConnector` by a `CanPhysicalChannel` that aggregates the aforementioned `CanFrameTriggering` – shall aggregate at least one of

- `CanControllerConfiguration` with `CanControllerXlConfiguration` aggregated or

- `CanControllerConfigurationRequirements` with `CanControllerXlConfigurationRequirements` aggregated.

⌋*()*

[constr_3705] Allowed values for `priorityId` ⌈The value for `priorityId` shall be in the range between 0 and 2047.

⌋*()*

[constr_3706] Allowed values for `sduType` ⌈The value for `sduType` shall be in the range between 0 and 255.

⌋*()*

[constr_3707] Allowed values for `vcid` ⌈The value for `vcid` shall be in the range between 0 and 255.

⌋*()*

[constr_3708] No UDP network management in case of Ethernet tunneling through CAN XL ⌈For an `EthernetPhysicalChannel` that is connected to an `EthernetCommunicationController` of category `CAN_XL` (i.e. an `EthernetPhysicalChannel` tunneled through CAN XL), no UDP network management shall be configured.

⌋*()*

[constr_3713] Allowed values for `acceptanceField` ⌈The value for `acceptanceField` shall be in the range between 0 and 4294967295.

⌋*()*

[constr_3714] Only one top level PNC-Coordinator per PNC ⌈In a `System` there shall be only up to one top level PNC-Coordinator defined per PNC according to [TPS_-SYST_03082].

⌋*()*

[constr_3716] `SecuredIPdu.dynamicRuntimeLengthHandling` for dynamic length `Pdu`s ⌈If a `PduTriggering` is referenced from a `SecuredIPdu` in the role `payload` and the `Pdu` referenced by the `PduTriggering` in the role `iPdu` has the attribute `Pdu.hasDynamicLength` set to true OR qualifies according to [TPS_SYST_-03085] to be of dynamic length, then the `SecuredIPdu` shall have the attribute `SecuredIPdu.dynamicRuntimeLengthHandling` set to true.

⌋*()*

[constr_3717] `SecuredIPdu.dynamicRuntimeLengthHandling` for gateway operation with `IPduMapping.pduMaxLength` defined ⌈If a `PduTriggering` refers to a `SecuredIPdu` in the role `iPdu` and that `PduTriggering` is used in an `IPduMapping` where a `pduMaxLength` value is defined (either in the role `IPduMapping.sourceIPdu` or `TargetIPduRef.targetIPdu`), then the `SecuredIPdu` shall have the attribute `SecuredIPdu.dynamicRuntimeLengthHandling` set to true.

⌋*()*

[constr_3718] Minimum length of `SecuredIPdu`s ⌈If a `SecuredIPdu` has the attribute `useAsCryptographicIPdu` set to false, then the `length` attribute of that `SecuredIPdu` shall be at least the sum of the `payload Pdu.length` and `SecuredIPdu.authenticationProps.authInfoTxLength`.

⌋*()*

[constr_3726]{DRAFT} **Upper multiplicity of aggregation in the role `CouplingPort.macSecProps`** ⌈In the context of `CouplingPort`, the aggregation in the role `macSecProps` shall exist at most once.

⌋*()*

[constr_4000] Local communication of mode switches ⌈Ports with `ModeSwitchInterface`s cannot be connected across ECU boundaries.

⌋*()*

[constr_5029] `J1939NmCluster` is not allowed to reference a `TtcanCluster` ⌈A `J1939NmCluster` is not allowed to reference a `TtcanCluster` in the role `communicationCluster`.

⌋*()*

[constr_5030] Uniqueness of `LinOrderedConfigurableFrame.index` ⌈`LinOrderedConfigurableFrame.index` shall always be set and be unique in the context of the aggregating `LinCommunicationConnector`.

⌋*()*

[constr_5031] Uniqueness of `FramePid.index` ⌈`FramePid.index` shall always be set and be unique in the context of the aggregating `AssignFrameIdRange`.

⌋*()*

[constr_5032] Maximal one `NmConfig` per `System` is allowed to be defined ⌈Each `System` element is allowed to reference at most one `NmConfig` element with the `fibexElement` reference.

⌋*()*

[constr_5049] Ethernet switch packet to traffic class assignment restriction ⌈For one `CouplingPortDetails` there exists either

- one `ethernetTrafficClassAssignment` with no `priority` attribute or
- up to 8 `ethernetTrafficClassAssignment` elements with a set of `priority` attributes

⌋*()*

[constr_5050] VariableDataPrototype of COM Based Transformer ⌈The `VariableDataPrototype` of [TPS_SYST_02058] shall be typed by an `ApplicationRecordDataType` or an `ImplementationDataType` of `category` STRUCTURE.

⌋*()*

[constr_5051] Existence of `CanFrameTriggering.identifier` in case of bus mirror target ⌈The `CanFrameTriggering` of a `Frame` that contains a `Pdu` of which

the `PduTriggering` is referenced by `BusMirrorChannelMappingCan` in the role `targetPduTriggering` shall not define an `identifier`.

⌋*()*

[constr_5053] Existence of `ISignalPort.handleInvalid` ⌈If the `ISignalPort` has a `networkRepresentationProps.invalidValue` defined then the `ISignalPort.communicationDirection` shall equal `in`.

⌋*()*

[constr_5054] `externalReplacement` not applicable for `ISignalPort.handleInvalid` ⌈In the context of `ISignalPort.handleInvalid` the value `externalReplacement` shall not be used.

⌋*()*

[constr_5055] `DataMapping` of elements of `PRPortPrototype`s is not supported ⌈A `DataMapping` shall not map elements of `PRPortPrototype`s to `SystemSignal`s

⌋*()*

[constr_5058] Value range for `CryptoServiceQueue.queueSize` ⌈If the `CryptoServiceQueue.queueSize` is defined it shall have a value which is equal or greater than 1.

⌋*()*

[constr_5060] Mapping of a `SecuredIPdu` into a `LinFrame` is not allowed ⌈The mapping of a `SecuredIPdu` into a `LinFrame` with a `PduToFrameMapping` is not allowed.

⌋*()*

[constr_5061] `EthernetCommunicationConnector`s and referencing `SocketAddress`es shall be in the same VLAN ⌈Each `EthernetCommunicationConnector` that is referenced by a `SocketAddress` in the role `connector` or `multicastConnector` shall be referenced by the same `EthernetPhysicalChannel` that aggregates the `SoAdConfig` that in turn aggregates the `SocketAddress`.

⌋*()*

[constr_5062] SOME/IP `ProvidedServiceInstance`s of the same serviceInterface on one `EcuInstance` ⌈Different `ProvidedServiceInstance`s with the same `serviceIdentifier` and the same `majorVersion` and different `instanceIdentifier`s shall not be mapped to the same UDP/TCP port number and IP address combination that is represented by referenced `ApplicationEndpoint` and its referenced `NetworkEndpoint`.

⌋*()*

[constr_5063] `ProvidedServiceInstance.serviceIdentifier` is mandatory
⌈The `ProvidedServiceInstance.serviceIdentifier` is mandatory.

⌋*()*

[constr_5064] `ProvidedServiceInstance.majorVersion` is mandatory ⌈The `ProvidedServiceInstance.majorVersion` is mandatory.

⌋*()*

[constr_5065] `ProvidedServiceInstance.minorVersion` is mandatory ⌈The `ProvidedServiceInstance.minorVersion` is mandatory.

⌋*()*

[constr_5066] `ProvidedServiceInstance.instanceIdentifier` is mandatory ⌈The `ProvidedServiceInstance.instanceIdentifier` is mandatory.

⌋*()*

[constr_5067] `ProvidedServiceInstance` shall be unique in respect of `serviceIdentifier`, `instanceIdentifier`, `majorVersion` ⌈On a VLAN each `ProvidedServiceInstance` shall have a different `serviceIdentifier`, `instanceIdentifier` and `majorVersion` value combination.

⌋*()*

[constr_5068] `ProvidedServiceInstance.localUnicastAddress` shall be IP Unicast ⌈If defined, the `ProvidedServiceInstance.localUnicastAddress` shall point to an IP Unicast address.

⌋*()*

[constr_5069] `ProvidedServiceInstance.remoteUnicastAddress` shall be IP Unicast ⌈The `ProvidedServiceInstance.remoteUnicastAddress` shall point to an IP Unicast address.

⌋*()*

[constr_5071] `EventHandler.eventMulticastAddress` reference target ⌈The `ApplicationEndpoint` that is referenced by an `EventHandler` in the role `eventMulticastAddress` shall reference a `NetworkEndpoint` that defines an IP Multicast Address.

⌋*()*

[constr_5072] `EventHandler` without defined `eventMulticastAddress` ⌈If an `EventHandler` that is aggregated by a `ProvidedServiceInstance` does not have a defined `eventMulticastAddress` then the `multicastThreshold` shall be set to the value 0 (IP Unicast only).

⌋*()*

[constr_5073] `PduActivationRoutingGroup` **with** `eventGroupControlType` **set to** `activationUnicast` **or** `triggerUnicast` **or** `activationAndTriggerUnicast` **that is aggregated by an** `EventHandler` ⌈An `EventHandler` that aggregates a `PduActivationRoutingGroup` with the `PduActivationRoutingGroup.eventGroupControlType` set to `activationUnicast` or `triggerUnicast` or `activationAndTriggerUnicast` shall be aggregated by a `ProvidedServiceInstance` that has a `localUnicastAddress` reference that points to an IP Unicast Address.

⌋*()*

[constr_5074] `PduActivationRoutingGroup` **with** `eventGroupControlType` **set to** `activationMulticast` **that is aggregated by an** `EventHandler` ⌈An `EventHandler` that aggregates a `PduActivationRoutingGroup` with the `PduActivationRoutingGroup.eventGroupControlType` set to `activationMulticast` shall have an `eventMulticastAddress` reference that points to a "remote" IP Multicast Address. The `ProvidedServiceInstance` that aggregates the `EventHandler` shall have a `localUnicastAddress` reference to a "local" UDP `ApplicationEndpoint`.

⌋*()*

[constr_5075] Allowed references of `SoConIPduIdentifier`**s by** `PduActivationRoutingGroup` **with** `eventGroupControlType` **set to** `activationMulticast` **and allowed** `SoConIPduIdentifier` **references** ⌈A `PduActivationRoutingGroup` with `eventGroupControlType` set to `activationMulticast` is allowed to reference `SoConIPduIdentifier`s only in the `iPduIdentifierUdp` role.

⌋*()*

[constr_5076] `PduActivationRoutingGroup` **with** `iPduIdentifierTcp` **reference that is aggregated by a** `ProvidedServiceInstance` ⌈If the `PduActivationRoutingGroup` contains the `iPduIdentifierTcp` reference then the aggregating `ProvidedServiceInstance` shall contain a `localUnicastAddress` reference to an `ApplicationEndpoint` that defines a TCP address.

⌋*()*

[constr_5077] `PduActivationRoutingGroup` **with** `iPduIdentifierUdp` **reference that is aggregated by a** `ProvidedServiceInstance` ⌈If the `PduActivationRoutingGroup` contains the `iPduIdentifierUdp` reference then the aggregating `ProvidedServiceInstance` shall contain a `localUnicastAddress` reference to an `ApplicationEndpoint` that defines a UDP address.

⌋*()*

[constr_5078] `PduTriggering`**s referenced by a** `PduActivationRoutingGroup` **shall be on the same VLAN as the referencing** `PduActivationRoutingGroup` ⌈Each `PduTriggering` referenced by a `PduActivationRoutingGroup` via `SoConIPduIdentifier` shall be aggregated by the same VLAN (`EthernetPhysi-`

calChannel) to which the `AbstractServiceInstance` that aggregates the `Pdu-ActivationRoutingGroup` belongs via the `localUnicastAddress`.

⌋*()*

[constr_5079] Service communication is restricted to one VLAN ⌈All `SocketAddress` elements that are referenced by a `AbstractServiceInstance` with the `localUnicastAddress` and `remoteUnicastAddress` shall belong to the same VLAN (`EthernetPhysicalChannel`).

⌋*()*

[constr_5080] `ApplicationEndpoint`s referenced by `EventHandler`s and by the aggregating `ProvidedServiceInstance` shall be in the same VLAN ⌈The `ApplicationEndpoint` that is referenced by an `EventHandler` in the role `eventMulticastAddress` shall belong to the same VLAN (`EthernetPhysicalChannel`) as the `ApplicationEndpoint` that is referenced by the `localUnicastAddress` reference from the `ProvidedServiceInstance` that aggregates the `EventHandler`.

⌋*()*

[constr_5081] `ConsumedServiceInstance.serviceIdentifier` is mandatory ⌈The `ConsumedServiceInstance.serviceIdentifier` is mandatory.

⌋*()*

[constr_5082] `ConsumedServiceInstance.majorVersion` is mandatory ⌈The `ConsumedServiceInstance.majorVersion` is mandatory.

⌋*()*

[constr_5083] `ConsumedServiceInstance.minorVersion` is mandatory ⌈The `ConsumedServiceInstance.minorVersion` is mandatory.

⌋*()*

[constr_5084] `ConsumedServiceInstance.instanceIdentifier` is mandatory ⌈The `ConsumedServiceInstance.instanceIdentifier` is mandatory.

⌋*()*

[constr_5085] `ConsumedServiceInstance.localUnicastAddress` shall be IP Unicast ⌈If defined, the `ConsumedServiceInstance.localUnicastAddress` shall point to an IP Unicast address.

⌋*()*

[constr_5086] `ConsumedServiceInstance.remoteUnicastAddress` shall be IP Unicast ⌈The `ConsumedServiceInstance.remoteUnicastAddress` shall point to an IP Unicast address.

⌋*()*

[constr_5087] **PduActivationRoutingGroup** with **eventGroupControlType** set to **activationUnicast** or **triggerUnicast** or **activationAndTriggerUnicast** that is referenced by a **ConsumedEventGroup** ⌈A ConsumedEvent-Group that aggregates a PduActivationRoutingGroup with the PduActivationRoutingGroup.eventGroupControlType set to activationUnicast or triggerUnicast or activationAndTriggerUnicast shall be aggregated by a ConsumedServiceInstance that has a localUnicastAddress reference that points to an IP Unicast Address.

⌋*()*

[constr_5088] **PduActivationRoutingGroup** with **iPduIdentifierTcp** reference that is aggregated by a **ConsumedServiceInstance** ⌈If the PduActivationRoutingGroup contains the iPduIdentifierTcp reference then the aggregating ConsumedServiceInstance shall contain a localUnicastAddress reference to an ApplicationEndpoint that defines a TCP address.

⌋*()*

[constr_5089] **PduActivationRoutingGroup** with **iPduIdentifierUdp** reference that is aggregated by a **ConsumedServiceInstance** ⌈If the PduActivationRoutingGroup contains the iPduIdentifierUdp reference then the aggregating ConsumedServiceInstance shall contain a localUnicastAddress reference to an ApplicationEndpoint that defines a UDP address.

⌋*()*

[constr_5090] **ApplicationEndpoint**s referenced by **ConsumedEventGroup**s and by the aggregating **ConsumedServiceInstance** shall be in the same VLAN ⌈The ApplicationEndpoint that is referenced by an ConsumedEventGroup in the role eventMulticastAddress shall belong to the same VLAN (EthernetPhysicalChannel) as the ApplicationEndpoint that is referenced by the localUnicastAddress reference from the ConsumedServiceInstance that aggregates the ConsumedEventGroup.

⌋*()*

[constr_5091] Relevance of **tcpRole** attribute ⌈The attribute tcpRole is only relevant if the StaticSocketConnection is aggregated by a SocketAddress that defines a TCP Port in the aggregated ApplicationEndpoint.

⌋*()*

[constr_5092] Local and remoteAddress of a StaticSocketConnection shall define the same transport protocol ⌈The transport protocol that is defined by the SocketAddress that aggregates the StaticSocketConnection shall be the same in the SocketAddress that is referenced by the same StaticSocketConnection in the role remoteAddress.

⌋*()*

[constr_5093] `pncGatewayType` **and** `PhysicalChannel` ⌈When multiple `CommunicationConnector`s with `pncGatewayType` set to a value other than `none` are referenced by the same `PhysicalChannel` then only up to one `CommunicationConnector` shall have the `pncGatewayType` set to `active`.

⌋*()*

[constr_5094] `pncGatewayType` **and ECU** ⌈When an ECU is connected to more than one `PhysicalChannel` and has a relation to a Partial Network then all `CommunicationConnector`s of this ECU where this Partial Network is related to shall have the `pncGatewayType` value either set to `none` or to a value different than none (i.e. `active` or `passive`).

⌋*()*

[constr_5095] Relationship between the timing behavior of the `ConsumedEventGroup` **retry and the timing behavior of an Offer message** ⌈The timing behavior for a retry to a `ConsumedEventGroup` (`subscribeEventgroupRetryMax`, `subscribeEventgroupRetryDelay`) shall not overlap to the timing behavior (`SomeipSdServerServiceInstanceConfig.offerCyclicDelay`) of the corresponding `ProvidedServiceInstance`.

⌋*()*

[constr_5096] `ConsumedEventGroup` **with value** `subscribeEventgroupRetryMax` **set to 255** ⌈Retry to a `ConsumedEventGroup` with value `subscribeEventgroupRetryMax` set to 255 is only allowed if the `SomeipSdServerServiceInstanceConfig.offerCyclicDelay` is set 0 and `serviceOfferTimeToLive` is set to 0xffffff of the corresponding `ProvidedServiceInstance`.

⌋*()*

[constr_5097] `DltLogChannel.txPduTriggering` **and** `DltLogChannel.rxPduTriggering` **shall point to** `GeneralPurposeIPdu`**s of category DLT** ⌈`DltLogChannel` shall only reference `PduTriggering`s that are pointing to `GeneralPurposeIPdu`s of category DLT in the roles `txPduTriggering` and `rxPduTriggering`.

⌋*()*

[constr_5100] Compatibility of two `MetaDataItemSet`**s** ⌈Under the condition that sender and receiver typed by a `SenderReceiverInterface` use meta-data and are mapped to the same `EcuInstance` the following condition applies: two `MetaDataItemSet`s are compatible if all of the following conditions are fulfilled:

- They aggregate the same number of `MetaDataItem`s.

- The value of `MetaDataItem.length` of corresponding `MetaDataItem`s is identical.

- The value of `MetaDataItem.metaDataItemType` of corresponding `MetaDataItem`s is identical.

⌋*()*

[constr_5101] Consistent Definition of meta-data ⌈If the `dataElement` referenced by a `SenderReceiverToSignalMapping` is also referenced by a `MetaDataItemSet` in the role `dataElement` and the mapping via `SystemSignal`, `ISignal`, and `ISignalToIPduMapping` down to an `ISignalIPdu` exists then all other `dataElement`s that are also mapped to the same `ISignalIPdu` shall either

- not be referenced by a `MetaDataItemSet` in the role `dataElement` (i.e. does not make use of meta-data) or

- the definition of meta-data in the context of the affected `SenderReceiverInterface`s is compatible (according to the definition of compatible specification of meta-data described in [constr_5100]).

⌋*()*

[constr_5104] Assignment of a `FlexrayFrame` where `allowDynamicLSduLength` is set to true ⌈`FlexrayFrame`s which are referenced by a `FlexrayFrameTriggering` where `allowDynamicLSduLength` is set to true shall always be assigned to the dynamic segment.

⌋*()*

[constr_5105] Mapping of `Pdu` with dynamic length in a `FlexrayFrame` ⌈Only the last `Pdu` in a `FlexrayFrame` is allowed to be a `Pdu` with `hasDynamicLength` = true.

⌋*()*

[constr_5106] `ISignalGroup` and `ISignal` referenced from `ISignalTriggering` ⌈Either an `ISignalGroup` and all `ISignal`s referenced from the `ISignalGroup` are also referenced from `ISignalTriggering`s aggregated at the same `PhysicalChannel` or neither the `ISignalGroup` nor any of the `ISignal`s referenced by the `ISignalGroup` shall be referenced from `ISignalTriggering`s.

⌋*()*

[constr_5110] Search for a collection of ServiceInstances is not supported ⌈The `ConsumedServiceInstance.instanceIdentifier` is not allowed to be set to the value ANY or ALL.

⌋*()*

[constr_5111] Existence of references `TlvDataIdDefinition.tlvArgument`, `TlvDataIdDefinition.tlvRecordElement`, and `TlvDataIdDefinition.tlvImplementationDataTypeElement` ⌈For each `TlvDataIdDefinition`, only one out of the following references shall exist:

- reference to `ArgumentDataPrototype` in the role `tlvArgument`

- reference to `ApplicationRecordElement` in the role `tlvRecordElement`

- reference to `ImplementationDataTypeElement` in the role `tlvImplementationDataTypeElement`.

⌋*()*

[constr_5112] `ImplementationDataType` needs to be defined if a "new-world" variable-size `ApplicationArrayDataType` is mapped to a single `SystemSignal` ⌈A `SenderReceiverInterface`.`dataElement` that is typed by a "new-world" variable-size `ApplicationArrayDataType` according to [TPS_SWCT_01644] (see definition in Software Component Template [2]) is only allowed to be mapped to a single `SystemSignal` by the `SenderReceiverToSignalMapping` if a `DataTypeMap` exists that points to both the `ApplicationArrayDataType` and an `ImplementationDataType` that fulfills the conditions of a "new-world" dynamic size array data type according to [TPS_SWCT_01645] (see definition in Software Component Template [2]).

⌋*()*

[constr_5113] Mapping of "old-world" variable size arrays to a single `SystemSignal` is not supported. ⌈The `SenderReceiverToSignalMapping` is not allowed to map a `dataElement` that is typed by an "old-world" variable size array defined by [TPS_SWCT_01641] and [TPS_SWCT_01642] (see definition in Software Component Template [2]) to a single `SystemSignal`.

⌋*()*

[constr_5114] Semantics of `InterpolationRoutine.isDefault` ⌈For each `SwRecordLayout` that is referenced by one or more `InterpolationRoutineMapping`s that are aggregated by `InterpolationRoutineMappingSet`s that are referenced from a `System` in the role `interpolationRoutineMappingSet`, only one of the collection of aggregated `InterpolationRoutine`s shall have attribute `isDefault` set to True.

⌋*()*

[constr_5116] Uniqueness of the symbols of software-components and BSW modules ⌈For all `SwComponentPrototype`s typed by an `ApplicationSwComponentType`, `NvBlockSwComponentType` or `SensorActuatorSwComponentType` mapped to a given `EcuInstance` by means of `SwcToEcuMapping` respectively `SwcToApplicationPartitionMapping` and `ApplicationPartitionToEcuPartitionMapping` the following restriction applies:

The symbolic name of an `AtomicSwComponentType` referenced by a respective `SwComponentPrototype` in the role `type` shall not overlap with the module implementation prefix (MIP) of any of the basic software-modules existing on the `EcuInstance`.

The symbolic name of an `AtomicSwComponentType` is derived from the value of

- `AtomicSwComponentType`.`symbol`, or if this attribute does not exist
- `AtomicSwComponentType`.`shortName`.

⌋*()*

[constr_5117] Client-Server communication over Ethernet ⌈A `SystemSignal` that is referenced by a `ClientServerToSignalMapping` in the role `callSignal` or `returnSignal` shall only be referenced by an `ISignal` that in turn is referenced by an `ISignalTriggering` aggregated by an `EthernetPhysicalChannel`.

⌋*()*

[constr_5118] Value range of `UdpProps.udpTtl` ⌈If defined, the value of `UdpProps.udpTtl` shall be in the range of 1..255.

⌋*()*

[constr_5119] Value range of `TcpProps.tcpTtl` ⌈If defined, the value of `TcpProps.tcpTtl` shall be in the range of 1..255.

⌋*()*

[constr_5120] Value range of `TcpProps.tcpDelayedAckTimeout` ⌈If defined, the value of `TcpProps.tcpDelayedAckTimeout` shall be in the range of 0..0.5.

⌋*()*

[constr_5121] Value range of `TcpProps.tcpSynMaxRtx` ⌈If defined, the value of `TcpProps.tcpSynMaxRtx` shall be in the range of 0..255.

⌋*()*

[constr_5122] Value range of `TcpProps.tcpMaxRtx` ⌈If defined, the value of `TcpProps.tcpMaxRtx` shall be in the range of 0..255.

⌋*()*

[constr_5123] Value range of `TcpProps.tcpKeepAliveProbesMax` ⌈If defined, the value of `TcpProps.tcpKeepAliveProbesMax` shall be in the range of 0..65535.

⌋*()*

[constr_5124] Value range of `TcpProps.tcpReceiveWindowMax` ⌈If defined, the value of `TcpProps.tcpReceiveWindowMax` shall be in the range of 0..65535.

⌋*()*

[constr_5125] Value range of `TcpIpIcmpv4Props.tcpIpIcmpV4Ttl` ⌈If defined, the value of `TcpIpIcmpv4Props.tcpIpIcmpV4Ttl` shall be in the range of 1..255.

⌋*()*

[constr_5126] Value range of `Ipv4ArpProps.tcpIpArpNumGratuitousArpOnStartup` ⌈If defined, the value of `Ipv4ArpProps.tcpIpArpNumGratuitousArpOnStartup` shall be in the range of 0..255.

⌋*()*

[constr_5127] **Value range of `Ipv4FragmentationProps.tcpIpIpNumFragments`** ⌈If defined, the value of `Ipv4FragmentationProps.tcpIpIpNumFragments` shall be in the range of 0..255.

⌋*()*

[constr_5128] **Value range of `Ipv4FragmentationProps.tcpIpIpNumReassDgrams`** ⌈If defined, the value of `Ipv4FragmentationProps.tcpIpIpNumReassDgrams` shall be in the range of 0..65535.

⌋*()*

[constr_5129] **Value range of `Ipv6FragmentationProps.tcpIpIpReassemblyBufferCount`** ⌈If defined, the value of `Ipv6FragmentationProps.tcpIpIpReassemblyBufferCount` shall be in the range of 0..255.

⌋*()*

[constr_5130] **Value range of `Ipv6FragmentationProps.tcpIpIpReassemblyBufferSize`** ⌈If defined, the value of `Ipv6FragmentationProps.tcpIpIpReassemblyBufferSize` shall be in the range of 1500..65535.

⌋*()*

[constr_5131] **Value range of `Ipv6FragmentationProps.tcpIpIpReassemblyTimeout`** ⌈If defined, the value of `Ipv6FragmentationProps.tcpIpIpReassemblyTimeout` shall be in the range of 0.001..100.

⌋*()*

[constr_5132] **Value range of `Ipv6FragmentationProps.tcpIpIpReassemblySegmentCount`** ⌈If defined, the value of `Ipv6FragmentationProps.tcpIpIpReassemblySegmentCount` shall be in the range of 1..255.

⌋*()*

[constr_5133] **Value range of `Ipv6FragmentationProps.tcpIpIpTxFragmentBufferCount`** ⌈If defined, the value of `Ipv6FragmentationProps.tcpIpIpTxFragmentBufferCount` shall be in the range of 1..1000.

⌋*()*

[constr_5134] **Value range of `Ipv6FragmentationProps.tcpIpIpTxFragmentBufferSize`** ⌈If defined, the value of `Ipv6FragmentationProps.tcpIpIpTxFragmentBufferSize` shall be in the range of 1500..65535.

⌋*()*

[constr_5135] **Value range of `Dhcpv6Props.tcpIpDhcpV6CnfDelayMin` and `Dhcpv6Props.tcpIpDhcpV6CnfDelayMax`** ⌈If defined, the value of `Dhcpv6Props.tcpIpDhcpV6CnfDelayMin` and the value of `Dhcpv6Props.tcpIpDhcpV6CnfDelayMax` shall be in the range of 0..100 and the value of

`Dhcpv6Props.tcpIpDhcpV6CnfDelayMax` shall be greater than the value of `Dhcpv6Props.tcpIpDhcpV6CnfDelayMin`.

⌋*()*

[constr_5136] **Value range of `Dhcpv6Props.tcpIpDhcpV6InfDelayMin` and `Dhcpv6Props.tcpIpDhcpV6InfDelayMax`** ⌈If defined, the value of `Dhcpv6Props.tcpIpDhcpV6InfDelayMin` and the value of `Dhcpv6Props.tcpIpDhcpV6InfDelayMax` shall be in the range of 0..100 and the value of `Dhcpv6Props.tcpIpDhcpV6InfDelayMax` shall be greater than the value of `Dhcpv6Props.tcpIpDhcpV6InfDelayMin`.

⌋*()*

[constr_5137] **Value range of `Dhcpv6Props.tcpIpDhcpV6SolDelayMin` and `Dhcpv6Props.tcpIpDhcpV6SolDelayMax`** ⌈If defined, the value of `Dhcpv6Props.tcpIpDhcpV6SolDelayMin` and the value of `Dhcpv6Props.tcpIpDhcpV6SolDelayMax` shall be in the range of 0..100 and the value of `Dhcpv6Props.tcpIpDhcpV6SolDelayMax` shall be greater than the value of `Dhcpv6Props.tcpIpDhcpV6SolDelayMin`.

⌋*()*

[constr_5138] **Value range of `Ipv6NdpProps.tcpIpNdpSlaacDadNumberOfTransmissions`** ⌈If defined, the value of `Ipv6NdpProps.tcpIpNdpSlaacDadNumberOfTransmissions` shall be in the range of 0..254.

⌋*()*

[constr_5139] **Value range of `Ipv6NdpProps.tcpIpNdpSlaacDadRetransmissionDelay`** ⌈If defined, the value of `Ipv6NdpProps.tcpIpNdpSlaacDadRetransmissionDelay` shall be in the range of 0..10.

⌋*()*

[constr_5140] **Value range of `Ipv6NdpProps.tcpIpNdpDefaultReachableTime`** ⌈If defined, the value of `Ipv6NdpProps.tcpIpNdpDefaultReachableTime` shall be in the range of 0..120.

⌋*()*

[constr_5141] **Value range of `Ipv6NdpProps.tcpIpNdpDefaultRetransTimer`** ⌈If defined, the value of `Ipv6NdpProps.tcpIpNdpDefaultRetransTimer` shall be in the range of 0..60.

⌋*()*

[constr_5142] **Value range of `Ipv6NdpProps.tcpIpNdpNumUnicastSolicitations`** ⌈If defined, the value of `Ipv6NdpProps.tcpIpNdpNumUnicastSolicitations` shall be in the range of 0..255.

⌋*()*

[constr_5143] **Value range of `Ipv6NdpProps.tcpIpNdpNumMulticastSolici-tations`** ⌈If defined, the value of `Ipv6NdpProps.tcpIpNdpNumMulticastSolic-itations` shall be in the range of 0..255.

⌋*()*

[constr_5144] **Value range of `Ipv6NdpProps.tcpIpNdpDelayFirstProbeTime`** ⌈If defined, the value of `Ipv6NdpProps.tcpIpNdpDelayFirstProbeTime` shall be in the range of 0..60.

⌋*()*

[constr_5145] **Value range of `Ipv6NdpProps.tcpIpNdpMinRandomFactor`** ⌈If defined, the value of `Ipv6NdpProps.tcpIpNdpMinRandomFactor` shall be in the range of 0..100.

⌋*()*

[constr_5146] **Value range of `Ipv6NdpProps.tcpIpNdpMaxRandomFactor`** ⌈If defined, the value of `Ipv6NdpProps.tcpIpNdpMaxRandomFactor` shall be in the range of 0..100.

⌋*()*

[constr_5147] **Value range of `Ipv6NdpProps.tcpIpNdpDestinationCacheSize`** ⌈If defined, the value of `Ipv6NdpProps.tcpIpNdpDestinationCacheSize` shall be in the range of 1..254.

⌋*()*

[constr_5148] **Value range of `Ipv6NdpProps.tcpIpNdpPrefixListSize`** ⌈If defined, the value of `Ipv6NdpProps.tcpIpNdpPrefixListSize` shall be in the range of 1..254.

⌋*()*

[constr_5149] **Value range of `Ipv6NdpProps.tcpIpNdpDefaultRouterList-Size`** ⌈If defined, the value of `Ipv6NdpProps.tcpIpNdpDefaultRouterListSize` shall be in the range of 2..254.

⌋*()*

[constr_5151] **Value range of `Ipv6NdpProps.tcpIpNdpMaxRtrSolicitations`** ⌈If defined, the value of `Ipv6NdpProps.tcpIpNdpMaxRtrSolicitations` shall be in the range of 0..255.

⌋*()*

[constr_5152] **Value range of `Ipv6NdpProps.tcpIpNdpMaxRtrSolicitation-Delay`** ⌈If defined, the value of `Ipv6NdpProps.tcpIpNdpMaxRtrSolicitation-Delay` shall be in the range of 0.001..60.

⌋*()*

[constr_5153] Value range of `Ipv6NdpProps.tcpIpNdpRtrSolicitationInterval` ⌈If defined, the value of `Ipv6NdpProps.tcpIpNdpRtrSolicitationInterval` shall be in the range of 0.001..60.

⌋*()*

[constr_5154] Value range of `TcpIpIcmpv6Props.tcpIpIcmpV6HopLimit` ⌈If defined, the value of `TcpIpIcmpv6Props.tcpIpIcmpV6HopLimit` shall be in the range of 1..255.

⌋*()*

[constr_5157] Mixing of Point-To-Point and Multi-Drop is not allowed in a `CouplingPortConnection` ⌈The `CouplingPortConnection` is allowed to reference a `CouplingPort` either:

- in the role `firstPort` and/or `secondPort` or

- in the role `nodePort`

⌋*()*

[constr_5158] Usage of `plcaProps` only allowed on 10BASE-T1S networks ⌈A `CouplingPort` is allowed to aggregate `plcaProps` only if:

- the `CouplingPort.physicalLayerType` is set to 10BASE-T1S

- the `CouplingPort.macLayerType` is set to `xMII`

- the `CouplingPort` is referenced by a `CouplingPortConnection` with the `nodePort` reference.

⌋*()*

[constr_5159] Mandatory `CouplingPortConnection` settings if multi-drop feature is used ⌈If a `CouplingPortConnection` uses the `nodePort` reference then the attribute `CouplingPortConnection.plcaLocalNodeCount` and the attribute `CouplingPortConnection.plcaTransmitOpportunityTimer` shall be set to a value.

⌋*()*

[constr_5160] Mandatory `PlcaProps` settings if multi-drop feature is used ⌈If a `CouplingPort` is referenced by a `CouplingPortConnection` in the role `nodePort` then the `CouplingPort` shall aggregate the `PlcaProps` and the following attributes shall be set to a value:

- `plcaMaxBurstCount`

- `plcaMaxBurstTimer`

- `plcaLocalNodeId`

⌋*()*

[constr_5162] **Valid** `TextTableMapping` **in the context of** `Sender-RecRecordElementMapping` ⌈The aggregation of a `TextTableMapping` at `SenderRecRecordElementMapping` is only valid if the `SenderRecRecordElementMapping` also references a `SystemSignal` in the role `systemSignal`.

⌋*()*

[constr_5163] Existence of attribute `IPSecRule.headerType` ⌈For each `IPSecRule`, the attribute `headerType` shall exist at the time when the COM Stack is generated.

⌋*()*

[constr_5164] Existence of attribute `IPSecRule.ipProtocol` ⌈For each `IPSecRule`, the attribute `ipProtocol` shall exist at the time when the COM Stack is generated.

⌋*()*

[constr_5165] Existence of attribute `IPSecRule.policy` ⌈For each `IPSecRule`, the attribute `policy` shall exist at the time when the COM Stack is generated.

⌋*()*

[constr_5166] Existence of `IPduMapping.pduMaxLength` ⌈If several `IPduMapping`s refer to the same `PduTriggering` in `IPduMapping.sourceIPdu`, then all of these `IPduMapping`s shall provide either no `IPduMapping.pduMaxLength` value, or the same `IPduMapping.pduMaxLength` value.

⌋*()*

[constr_5167] `pncGatewayType` **and ECU over the whole system** ⌈Only one PNC Gateway ECU in the whole System shall exist that sets on all its `CommunicationConnector`s the `pncGatewayType` to `active`.

⌋*()*

[constr_5168] pncGatewayType passive and connected ECUs ⌈For all `CommunicationConnector`s with `pncGatewayType` set to passive belonging to one PNC Gateway ECU, all connected counterpart `CommunicationConnector`s where `pncGatewayType` is set to active shall belong to one ECU.

⌋*()*

[constr_5169] pncGatewayType and (routing) paths ⌈No path over all networks shall exist that connects a `CommunicationConnector` with `pncGatewayType` active to a `CommunicationConnector` with `pncGatewayType` passive where both `CommunicationConnector`s belong to the same ECU.

⌋*()*

[constr_5170] `nmPassiveModeEnabled` **and** `dynamicPncToChannelMappingEnabled` ⌈If `nmPassiveModeEnabled` is set to TRUE on a `NmNode` then `dynam-`

`icPncToChannelMappingEnabled` shall be set to FALSE on the according `CommunicationConnector` referring to the same `CommunicationController`.

⌋*()*

[constr_5175]{DRAFT} **`RtePluginProps` shall reference at least one `EcucContainerValue` representing a RteRipsPlugin** ⌈If a `FlatInstanceDescriptor` owns are `RtePluginProps` this `RtePluginProps` shall define the `associatedRtePlugin` reference and/or the `associatedCrossSwClusterComRtePlugin` reference.

⌋*()*

[constr_5176]{DRAFT} **Existence of `CpSoftwareCluster` of `category` HOST_SOFTWARE_CLUSTER on one `EcuInstance`** ⌈On each `EcuInstance`, exactly one `CpSoftwareCluster` of `category` HOST_SOFTWARE_CLUSTER shall exist.

⌋*()*

[constr_5177]{DRAFT} **Validity of reference `CpSoftwareClusterToEcuInstanceMapping.swCluster`** ⌈A `CpSoftwareClusterToEcuInstanceMapping` that references a given `CpSoftwareCluster` in the role `CpSoftwareClusterToEcuInstanceMapping.swCluster` shall be aggregated by the same `System` (in the role `System.mapping.swMapping`) that also refers to the referenced `CpSoftwareCluster` in the role `System.swCluster`.

⌋*()*

[constr_5178]{DRAFT} **Existence of attribute `CpSoftwareClusterResource.globalResourceId`** ⌈For each `CpSoftwareClusterResource`, attribute `globalResourceId` shall exist **at the time when the definition of the resource pool is finished**.

⌋*()*

[constr_5179]{DRAFT} **Existence of attribute `CpSoftwareClusterResource.isMandatory`** ⌈For each `CpSoftwareClusterResource`, attribute `isMandatory` shall exist **at the time when the definition of the resource pool is finished**.

⌋*()*

[constr_5180]{DRAFT} **Allowed values for `CpSoftwareClusterResource.globalResourceId`** ⌈Attribute `CpSoftwareClusterResource.globalResourceId` shall not be set to 0.

⌋*()*

[constr_5181]{DRAFT} **Existence of attribute `CpSoftwareClusterServiceResource.category`** ⌈For each `CpSoftwareClusterServiceResource`, attribute

`category` shall exist **at the time when the definition of the resource pool is finished**.

⌋*()*

[constr_5182]{DRAFT} **PRPortPrototypes are excluded as CpSoftwareCluster interfaces** ⌈A `CpSoftwareClusterCommunicationResource` is not allowed to be mapped by a `PortElementToCommunicationResourceMapping` to an element of a `PortInterface` in the context of a `PRPortPrototype`.

⌋*()*

[constr_5183]{DRAFT} **PortElementToCommunicationResourceMapping shall reference exactly one element of a `PortInterface`** ⌈For any given `PortElementToCommunicationResourceMapping`, either the reference

- `parameterDataPrototype` or

- `modeDeclarationGroupPrototype` or

- `trigger` or

- `clientServerOperation` or

- `variableDataPrototype`

shall exist.

⌋*()*

[constr_5184]{DRAFT} **CpSoftwareClusterServiceResource can be provided only once on an `EcuInstance`** ⌈A `CpSoftwareClusterServiceResource` shall not be mapped by several `CpSoftwareClusterToResourceMapping`s to `CpSoftwareCluster`s in the `provider` role if the `CpSoftwareCluster`s are mapped to the same `EcuInstance` by `CpSoftwareClusterToEcuInstanceMapping`s.

⌋*()*

[constr_5185]{DRAFT} **Existence of attribute BinaryManifestProvideResource.globalResourceId** ⌈For each `BinaryManifestProvideResource`, attribute `globalResourceId` shall exist **at the time when the definition of binary object meta-data is finished**.

⌋*()*

[constr_5186]{DRAFT} **Existence of attribute BinaryManifestProvideResource.resourceGuardValue** ⌈For each `BinaryManifestProvideResource`, attribute `resourceGuardValue` shall exist **at the time when the definition of binary object meta-data is finished**.

⌋*()*

[constr_5187]{DRAFT} **Existence of attribute `BinaryManifestProvideResource.supportsMultipleNotifierSets`** ⌈For each `BinaryManifestProvideResource`, attribute `supportsMultipleNotifierSets` shall exist **at the time when the definition of binary object meta-data is finished**.

⌋*()*

[constr_5188]{DRAFT} **Existence of attribute `BinaryManifestProvideResource.numberOfNotifierSets`** ⌈For each `BinaryManifestProvideResource`, attribute `numberOfNotifierSets` shall exist **at the time when the definition of binary object meta-data is finished**.

⌋*()*

[constr_5189]{DRAFT} **Existence of reference `BinaryManifestProvideResource.resourceDefinition`** ⌈For each `BinaryManifestProvideResource`, the reference in the role `resourceDefinition` shall exist **at the time when the definition of binary object meta-data is finished**.

⌋*()*

[constr_5190]{DRAFT} **Existence of aggregation `BinaryManifestProvideResource.item`** ⌈For each `BinaryManifestProvideResource`, the aggregation in the role `item` shall exist at least once **at the time when the definition of binary object meta-data is finished**.

⌋*()*

[constr_5191]{DRAFT} **Consequence of attribute `BinaryManifestProvideResource.item.category`** ⌈The following values of attribute `BinaryManifestProvideResource`.`item`.`category` shall require the existence of aggregations:

- If `category` is set to `PROVIDER_HANDLE` and the attribute `isUnused` is not set to true then the aggregation `BinaryManifestProvideResource`.`item`.`value` shall exist **at the time when the definition of binary object meta-data is finished**.

- If `category` is set to `NOTIFIER_HANDLE` and the attribute `isUnused` is not set to true then the aggregation `BinaryManifestProvideResource`.`item`.`defaultValue` shall exist **at the time when the definition of binary object meta-data is finished**.

- If `category` is set to `AUXILARY_ACTUAL_NUMBER_NOTIFIER_SETS` then the aggregation `BinaryManifestProvideResource`.`item`.`defaultValue` shall exist **at the time when the definition of binary object meta-data is finished**.

⌋*()*

[constr_5192]{DRAFT} **Existence of attribute `BinaryManifestRequireResource.globalResourceId`** ⌈For each `BinaryManifestRequireResource`, at-

tribute `globalResourceId` shall exist **at the time when the definition of binary object meta-data is finished**.

⌋*()*

[constr_5193]{DRAFT} **Existence of attribute `BinaryManifestRequireResource.resourceGuardValue`** ⌈For each `BinaryManifestRequireResource`, attribute `resourceGuardValue` shall exist **at the time when the definition of binary object meta-data is finished**.

⌋*()*

[constr_5194]{DRAFT} **Existence of reference `BinaryManifestRequireResource.resourceDefinition`** ⌈For each `BinaryManifestRequireResource`, the reference in the role `resourceDefinition` shall exist **at the time when the definition of binary object meta-data is finished**.

⌋*()*

[constr_5195]{DRAFT} **Existence of aggregation `BinaryManifestRequireResource.item`** ⌈For each `BinaryManifestRequireResource`, the aggregation in the role `item` shall exist at least once **at the time when the definition of binary object meta-data is finished**.

⌋*()*

[constr_5196]{DRAFT} **Consequence of attribute `BinaryManifestRequireResource.item.category`** ⌈The following values of attribute `BinaryManifestRequireResource.item.category` shall require the existence of aggregations:

- If `category` is set to `PROVIDER_HANDLE` then the aggregation `BinaryManifestRequireResource.item.defaultValue` shall exist **at the time when the definition of binary object meta-data is finished**.

- If `category` is set to `NOTIFIER_HANDLE` then the aggregation `BinaryManifestRequireResource.item.value` shall exist **at the time when the definition of binary object meta-data is finished**.

⌋*()*

[constr_5197]{DRAFT} **Existence of aggregation `BinaryManifestResourceDefinition.itemDefinition`** ⌈For each `BinaryManifestResourceDefinition`, the aggregation in the role `itemDefinition` shall exist at least once **at the time when the definition of binary object meta-data is finished**.

⌋*()*

[constr_5198]{DRAFT} **Allowed `BinaryManifestResource.resourceDefinition`** ⌈An `BinaryManifestResourceDefinition` shall only be referenced from a `BinaryManifestResource` that is aggregated in the same `CpSoft-`

`wareClusterBinaryManifestDescriptor` as the referenced `BinaryManifestResourceDefinition`.

⌋*()*

[constr_5199]{DRAFT} **Consequence of attribute `BinaryManifestItem.auxiliaryField.category`** ⌈If attribute `BinaryManifestItem.auxiliaryField.category` is set to value `AUXILARY_CONNECTED_SW_CLUSTER_ID` then attribute `BinaryManifestItem.auxiliaryField.defaultValue` shall exist **at the time when the definition of binary object meta-data is finished**.

⌋*()*

[constr_5200]{DRAFT} **Existence of attribute `BinaryManifestItemDefinition.category`** ⌈For each `BinaryManifestItemDefinition`, attribute `category` shall exist **at the time when the definition of binary object meta-data is finished**.

⌋*()*

[constr_5201]{DRAFT} **Existence of attribute `BinaryManifestItemDefinition.size`** ⌈For each `BinaryManifestItemDefinition`, attribute `size` shall exist **at the time when the definition of binary object meta-data is finished**.

⌋*()*

[constr_5202]{DRAFT} **Existence of attribute `BinaryManifestItemNumericalValue.value`** ⌈For each `BinaryManifestItemNumericalValue`, attribute `value` shall exist **at the time when the definition of binary object meta-data is finished**.

⌋*()*

[constr_5203]{DRAFT} **Existence of attribute `BinaryManifestItemPointerValue.symbol`** ⌈For each `BinaryManifestItemPointerValue`, attribute `symbol` shall exist **at the time when the definition of binary object meta-data is finished**.

⌋*()*

[constr_5204]{DRAFT} **Existence of attribute `BinaryManifestMetaDataField.category`** ⌈For each `BinaryManifestMetaDataField`, attribute `category` shall exist **at the time when the definition of binary object meta-data is finished**.

⌋*()*

[constr_5205]{DRAFT} **Existence of attribute `BinaryManifestMetaDataField.size`** ⌈For each `BinaryManifestMetaDataField`, attribute `size` shall exist **at the time when the definition of binary object meta-data is finished**.

⌋*()*

[constr_5206]{DRAFT} **Existence of attribute `BinaryManifestMetaDataField`.`symbol`** ⌈For each `BinaryManifestMetaDataField`, attribute `symbol` shall exist **at the time when the definition of binary object meta-data is finished**.

⌋*()*

[constr_5207]{DRAFT} **Existence of attribute `BinaryManifestMetaDataField`.`address`** ⌈For each `BinaryManifestMetaDataField`, attribute `address` shall exist **at the time when the definition of binary object meta-data is finished**.

⌋*()*

[constr_5208]{DRAFT} **Existence of `System.swCluster`** ⌈In a `System` with `category` SW_CLUSTER_SYSTEM_DESCRIPTION, the reference `System.swCluster` shall exist at least once **at the time when the software cluster extraction is finished**.

⌋*()*

[constr_5209]{DRAFT} **Existence of reference `CpSoftwareCluster.swComponentAssignment.swComponent`** ⌈In a `System` with `category` SW_CLUSTER_SYSTEM_DESCRIPTION, the reference `System.swCluster.swComponentAssignment.swComponent` shall exist **at the time when the software cluster extraction is finished**.

⌋*()*

[constr_5210]{DRAFT} **Existence of reference `SystemMapping.portElementToComResourceMapping`** ⌈In a `System` with `category` SW_CLUSTER_SYSTEM_DESCRIPTION, the reference `System.mapping.portElementToComResourceMapping` shall exist at least once **at the time when the software cluster extraction is finished**.

⌋*()*

[constr_5211]{DRAFT} **Existence of reference `PortElementToCommunicationResourceMapping.communicationResource`** ⌈In a `System` with `category` SW_CLUSTER_SYSTEM_DESCRIPTION, the reference `System.mapping.portElementToComResourceMapping.communicationResource` shall exist at least once **at the time when the software cluster extraction is finished**.

⌋*()*

[constr_5212]{DRAFT} **Existence of reference `SystemMapping.resourceToApplicationPartitionMapping`** ⌈In a `System` with `category` SW_CLUSTER_SYSTEM_DESCRIPTION, the reference `System.mapping.resourceToApplicationPartitionMapping` shall exist **at the time when the software cluster extraction is finished**.

⌋*()*

[constr_5213]{DRAFT} **Existence of reference `CpSoftwareClusterResource-ToApplicationPartitionMapping.applicationPartition`** ⌈In a `System` with `category` SW_CLUSTER_SYSTEM_DESCRIPTION, the reference `System.mapping.resourceToApplicationPartitionMapping.applicationPartition` shall exist **at the time when the software cluster extraction is finished**.

⌋*()*

[constr_5214]{DRAFT} **Existence of reference `CpSoftwareClusterResource-ToApplicationPartitionMapping.resource`** ⌈In a `System` with `category` SW_CLUSTER_SYSTEM_DESCRIPTION, the reference `System.mapping.resourceToApplicationPartitionMapping.resource` shall exist **at the time when the software cluster extraction is finished**.

⌋*()*

[constr_5215]{DRAFT} **Existence of reference `CpSoftwareClusterToResourceMapping.serviceResource`** ⌈In a `System` with `category` SW_CLUSTER_SYSTEM_DESCRIPTION, the reference `System.mapping.softwareClusterToResourceMapping.serviceResource` shall exist **at the time when the software cluster extraction is finished**.

⌋*()*

[constr_5216]{DRAFT} **Existence of reference `CpSoftwareClusterToResourceMapping.requester` and/or `provider`** ⌈In a `System` with `category` SW_CLUSTER_SYSTEM_DESCRIPTION, at least one of the references `System.mapping.softwareClusterToResourceMapping.requester` or `System.mapping.softwareClusterToResourceMapping.provider` shall exist **at the time when the software cluster extraction is finished**.

⌋*()*

[constr_5217]{DRAFT} **Existence of attribute `BinaryManifestMetaDataField.value`** ⌈For each `BinaryManifestMetaDataField` of `category` IMMUTABLE_TABLES_CHECKSUM, attribute `value` shall exist **at the time when the definition of binary object meta-data is finished**.

⌋*()*

[constr_5218]{DRAFT} **Existence of attribute `BinaryManifestItemPointerValue.address`** ⌈For each `BinaryManifestItemPointerValue`, attribute `address` shall exist **at the time when the definition of binary object meta-data is finished**.

⌋*()*

[constr_5219]{DRAFT} **`CpSoftwareCluster` shall only be mapped to one `EcuInstance`** ⌈Within the context of one `CpSoftwareCluster`, for all `CpSoftwareCluster.swComponentAssignment.swComponent` (and nested instances of `SwComponentPrototype`s) that are referenced by a `SwcToEcuMapping` in the role `component` the following condition shall be be fulfilled: all referencing `SwcToEcuMap-`

`ping`s shall refer to the same `EcuInstance` in the role `ecuInstance` and this `EcuInstance` shall also be referenced in the role `ecuInstance` by all `CpSoftwareClusterToEcuInstanceMapping`s that also refer to said `CpSoftwareCluster` in the role `swCluster`.

⌋*()*

[constr_5220] Multiplicity of `EndToEndTransformationISignalProps.sourceId` in PROFILE_04m, PROFILE_07m, PROFILE_08m and PROFILE_44m ⌈If the `EndToEndTransformationDescription.profileName` attribute is set to PROFILE_04m, PROFILE_07m, PROFILE_08m, or PROFILE_44m then the multiplicity of the `EndToEndTransformationISignalProps.sourceId` attribute shall be 1.

⌋*()*

[constr_5221] Multiplicity of `EndToEndTransformationISignalProps.sourceId` in PROFILE_01, PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_11, and PROFILE_22 ⌈If the `EndToEndTransformationDescription.profileName` attribute is set to PROFILE_01, PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_11, or PROFILE_22 then the multiplicity of the `EndToEndTransformationISignalProps.sourceId` attribute shall be 0.

⌋*()*

[constr_5222] Mandatory elements of `UdpNmCluster` ⌈The following attributes shall always be defined for the `UdpNmCluster`:

- `nmMsgCycleTime`
- `nmMessageTimeoutTime`
- `nmNetworkTimeout`
- `nmRemoteSleepIndicationTime`
- `nmRepeatMessageTime`
- `nmWaitBusSleepTime`
- `communicationCluster`

⌋*()*

[constr_5223] Mandatory elements of `UdpNmNode` ⌈The following attributes shall always be defined for the `UdpNmNode`:

- `nmMsgCycleOffset`

⌋*()*

[constr_5224] UdpNmNode.nmMsgCycleOffset < UdpNmCluster.nmMsgCycleTime ⌈The value of UdpNmNode.nmMsgCycleOffset shall be smaller than the value of UdpNmCluster.nmMsgCycleTime.

⌋*()*

[constr_5225] UdpNmCluster.nmNetworkTimeout multiple of UdpNmCluster.nmMsgCycleTime ⌈The value of UdpNmCluster.nmNetworkTimeout shall be n * UdpNmCluster.nmMsgCycleTime with n > 1.

⌋*()*

[constr_5226] UdpNmCluster.nmRepeatMessageTime multiple of UdpNmCluster.nmMsgCycleTime ⌈The value of UdpNmCluster.nmRepeatMessageTime shall be n * UdpNmCluster.nmMsgCycleTime.

⌋*()*

[constr_5229] Existence of attribute E2EProfileCompatibilityProps.transitToInvalidExtended is mandatory for each EndToEndTransformationDescription ⌈For each EndToEndTransformationDescription, a reference to E2EProfileCompatibilityProps in the role e2eProfileCompatibilityProps shall exist and the referenced E2EProfileCompatibilityProps shall define a value for the attribute transitToInvalidExtended.

⌋*()*

[constr_5231] Allowed values for SOMEIPTransformationProps.alignment and SOMEIPTransformationDescription.alignment ⌈The valid values for SOMEIPTransformationProps.alignment and SOMEIPTransformationDescription.alignment shall be 8, 16, 32, 64, 128 or 256.

⌋*()*

[constr_5232] Triggering in case of application writing the selector field signal ⌈If

- the ISignal representing the selector field is referenced by an ISignalTriggering and that ISignalTriggering refers to an ISignalPort where the communicationDirection is set to out and

- the ISignal representing the selector field is referring to a SystemSignal and that SystemSignal is either

  - referenced by a SenderReceiverToSignalMapping in the role systemSignal or

  - part of a SystemSignalGroup that in turn is referenced by a SenderReceiverToSignalGroupMapping

then any `ISignal` other than the `ISignal` representing the selector field shall be mapped into that dynamic part alternative `ISignalIPdu` using the `transferProperty` set to `pending`.

⌋*()*

[constr_5233] Usage of `invalidValue` in case of application writing the selector field signal ⌈If

- the `ISignal` representing the selector field is referenced by an `ISignalTriggering` and that `ISignalTriggering` refers to an `ISignalPort` where the `communicationDirection` is set to `out` and

- the `ISignal` representing the selector field is referring to a `SystemSignal` and that `SystemSignal` is either

  - referenced by a `SenderReceiverToSignalMapping` in the role `systemSignal` or

  - part of a `SystemSignalGroup` that in turn is referenced by a `SenderReceiverToSignalGroupMapping`

then

- the `ISignal` representing the selector field shall either

  - define no invalid value (`ISignal.networkRepresentationProps.invalidValue`) or

  - the `invalidValue` defined shall be different than any of the defined selector field values for that `MultiplexedIPdu`.

⌋*()*

[constr_5235] Maximum `Frame.frameLength` of the used bus protocol shall not be exceeded ⌈The `Pdu.length` used for an `IPdu` and the `IPduMapping.pduMaxLength` used for a `targetIPdu` shall not exceed the limitation of the maximum `Frame.frameLength` of the used bus protocol (e.g. CAN2.0 max. `Frame.frameLength` == 8Byte, CAN-FD `Frame.frameLength` == 64byte).

⌋*()*

[constr_5236] Restriction of `IPduMapping.pduMaxLength` ⌈`IPduMapping.pduMaxLength` shall be equal or greater than the maximum `Pdu.length` of `sourceIPdu` and `targetIPdu`. For a N:1 routing and 1:N routing, respectively, the maximum `Pdu.length` of all involved `Pdu`s shall be used to evaluate a proper `IPduMapping.pduMaxLength`.

⌋*()*

[constr_5244] Value of attribute `SOMEIPTransformationISignalProps.sizeOfArrayLengthFields` ⌈If attribute `SOMEIPTransformationISignalProps.sizeOfArrayLengthFields` is configured, then the value of attribute `SOMEIP`-

`TransformationISignalProps`.`sizeOfArrayLengthFields` shall be at least as high as the number of bytes required to fit the maximum result of the individual length field computation of all variable-size arrays that are transported in the SOME/IP message.

In other words, for each variable-size array contained in the SOME/IP message, the numerical value of *maximum number of elements * sizeof(data type of array element)* shall be computed which yields the maximum number of bytes required to store the individual variable-size array.

The size of the attribute `SOMEIPTransformationISignalProps`.`sizeOfArrayLengthFields` shall be set such that the highest value (or bigger) obtained from the individual computations for the contained variable-size arrays can fit into the length field. The unit of attribute `SOMEIPTransformationISignalProps`.`sizeOfArrayLengthFields` is bytes.

⌋*()*

[constr_5245] Value of attribute `SOMEIPTransformationISignalProps.sizeOfStringLengthFields` ⌈If attribute `SOMEIPTransformationISignalProps`.`sizeOfStringLengthFields` is configured, then the value of attribute `SOMEIPTransformationISignalProps`.`sizeOfStringLengthFields` shall be at least as high as the number of bytes required to fit the maximum result of the individual length field computation of all strings that are transported in the SOME/IP message.

In other words, for each string contained in the SOME/IP message, the numerical value of *maximum number of characters in the string * maximum number of code units per character (of the used character encoding) * maximum number of bytes per code unit (of the used character encoding)* shall be computed which yields the maximum number of bytes required to store the individual string.

The size of the attribute `SOMEIPTransformationISignalProps`.`sizeOfStringLengthFields` shall be set such that the highest value (or bigger) obtained from the individual computations for the contained strings can fit into the length field. The unit of attribute `SOMEIPTransformationISignalProps`.`sizeOfStringLengthFields` is bytes.

⌋*()*

[constr_5246] SOME/IP Transformation settings for strings in the context of an `ISignal` ⌈In the context of an `ISignal` the usage of `DataPrototypeTransformationProps`.`transformationProps`.`sizeOfStringLengthField` is only allowed if the `SOMEIPTransformationISignalProps`.`sizeOfStringLengthFields` is not defined.

⌋*()*

[constr_5247] Value of attribute `DataPrototypeTransformationProps.transformationProps.sizeOfArrayLengthField` ⌈If the configuration of length field is done using `DataPrototypeTransformationProps`.`transformationProps`.

`sizeOfArrayLengthField` then the value of attribute `DataPrototypeTransformationProps.transformationProps.sizeOfArrayLengthField` shall be at least as high as the number of bytes required to fit the result of the expression *maximum number of elements * sizeof(data type of array element)*.

⌋*()*

[constr_5248] Value of attribute `DataPrototypeTransformationProps.transformationProps.sizeOfStringLengthField` ⌈If the configuration of length field is done using `DataPrototypeTransformationProps.transformationProps.sizeOfStringLengthField` then the value of attribute `DataPrototypeTransformationProps.transformationProps.sizeOfStringLengthField` shall be at least as high as the number of bytes required to fit the result of the expression *maximum number of characters in the string * maximum number of code units per character (of the used character encoding) * maximum number of bytes per code unit (of the used character encoding)*.

⌋*()*

[constr_5249] Existence of `Pdu.length` ⌈For each `Pdu`, attribute `length` shall exist at the time when the Ecu configuration of the COM stack is created.

⌋*()*

[constr_5251] `CouplingPort.connectionNegotiationBehavior` shall exist ⌈The attribute `CouplingPort.connectionNegotiationBehavior` shall be defined at the time the Base EcuC is created.

⌋*()*

[constr_5252] `LinSlaveConfig.protocolVersion` shall exist ⌈The attribute `LinSlaveConfig.protocolVersion` shall be defined at the time the Base EcuC is created.

⌋*()*

[constr_5253] Value range of `ISignal.length` ⌈The value of `ISignal.length` shall be in the range of 0..34359738360 Bits.

⌋*()*

[constr_5254] Value range of `MultiplexedIPdu.selectorFieldLength` ⌈The value of `MultiplexedIPdu.selectorFieldLength` shall be in the range of 1..16 Bits.

⌋*()*

[constr_5258] `TriggerToSignalMapping.systemSignal`s eligible for a `TriggerToSignalMapping` in case `DataTransformation` is used ⌈The `ISignal` that is referenced by a `SystemSignal` that in turn is referenced by a `TriggerToSignalMapping` in the role `systemSignal` shall have its `length` attribute set to the value of `BufferProperties.headerLength` attribute of the respective `Transfor-`

mationTechnology if the `ISignal` references a `DataTransformation` in the role `dataTransformation` that in turn references the `TransformationTechnology`.

⌋*()*

[constr_5259] `PduTriggering`s and `FrameTriggering`s of `SecuredIPdu` with `useAsCryptographicIPdu` = true ⌈In case that a `SecuredIPdu` is defined with `useAsCryptographicIPdu` = true as described by [TPS_SYST_02173] then:

- the `PduTriggering` of the `AuthenticPdu`

- the `PduTriggering` of the `CryptographicPdu`

- the `FrameTriggering` that references the `Frame` to which the `AuthenticPdu` is mapped

- the `FrameTriggering` that references the `Frame` to which the `CryptographicPdu` is mapped

shall be aggregated by the same `PhysicalChannel`.

⌋*()*

[constr_5262] `SystemSignal` used for Trigger communication shall not be part of any `SystemSignalGroup` ⌈A `SystemSignal` that is target of a `TriggerToSignalMapping` in the role `systemSignal` shall not be referenced by a `SystemSignalGroup` in the role `systemSignal`.

⌋*()*

[constr_5263] `NetworkEndpoint.networkEndpointAddress` restriction for IPv4 ⌈A `NetworkEndpoint` shall not aggregate several `Ipv4Configuration`s that have their `ipv4AddressSource` set to fixed.

⌋*()*

[constr_5264] `NetworkEndpoint.networkEndpointAddress` restriction for IPv6 ⌈A `NetworkEndpoint` shall not aggregate several `Ipv6Configuration`s that have their `ipv6AddressSource` set to fixed.

⌋*()*

[constr_5265] `NetworkEndpoint.networkEndpointAddress` restriction ⌈A `NetworkEndpoint` shall not aggregate an `Ipv4Configuration` and an `Ipv6Configuration` as `networkEndpointAddress` at the same time.

⌋*()*

[constr_5266] `VariableDataPrototype` of `NvDataInterface` shall not be mapped to a `SystemSignal` ⌈A `VariableDataPrototype` that is aggregated by a `NvDataInterface` shall not be referenced by

- `SenderReceiverToSignalGroupMapping` in the role `dataElement` and

- `SenderReceiverToSignalMapping` in the role `dataElement`.

⌋*()*

[constr_5267] **`VariableDataPrototype` of `NvDataInterface` shall not be mapped to a `SystemSignal` via a delegation to a `PortPrototype` with a `SenderReceiverInterface`** ⌈If a `VariableDataPrototype` that is aggregated by a

- `SenderReceiverInterface` and that `SenderReceiverInterface` is referenced by a `PortPrototype` of a Composition and

- that `PortPrototype` is connected by a delegation connector with an inner `PortPrototype` of a `NvBlockSwComponentType` and

- that `PortPrototype` is typed by a `NvDataInterface`

then this `PortPrototype` shall not be referenced by:

- `SenderReceiverToSignalGroupMapping` in the role `dataElement` and

- `SenderReceiverToSignalMapping` in the role `dataElement`.

⌋*()*

[constr_5268] Existence of `ContainedIPduProps.containedPduTriggering` reference ⌈If a `ContainedIPduProps` is aggregated at the `ContainerIPdu` in the role `ContainerIPdu.containedIPduTriggeringProps` then the reference `ContainedIPduProps.containedPduTriggering` shall exist.

⌋*()*

[constr_5269] Exclusion of `ContainedIPduProps.containedPduTriggering` reference ⌈If a `ContainedIPduProps` is aggregated at the `IPdu` in the role `IPdu.containedIPduProps` then the reference `ContainedIPduProps.containedPduTriggering` shall NOT exist.

⌋*()*

[constr_5270] Exclusive usage of `ContainerIPdu.containedPduTriggering` and `ContainerIPdu.containedIPduTriggeringProps` ⌈A `ContainerIPdu` shall only have either `ContainerIPdu.containedPduTriggering` OR `ContainerIPdu.containedIPduTriggeringProps` defined.

⌋*()*

[constr_5271] Existence of attribute `BinaryManifestItem.isUnused` ⌈For each `BinaryManifestItem`, the attribute `isUnused` shall exist at the time when the definition of binary object meta-data is finished.

⌋*()*

[constr_5272] Value of attribute `BinaryManifestItem.isUnused` ⌈The attribute `BinaryManifestItem`.isUnused shall only permitted to be set to true if the related `BinaryManifestItemDefinition` has its attribute `isOptional` set to true,

⌋*()*

[constr_5273] One `ISignalTriggering` pair allowed per `EthernetPhysicalChannel` for a `ClientServerOperation` ⌈In the context of a `System` of category ECU_SYSTEM_DESCRIPTION or ECU_EXTRACT, for each `EthernetPhysicalChannel` at most one pair of

- `ISignalTriggering` that refers to an `ISignal` that in turn refers to a `SystemSignal` that is referenced by a specific `ClientServerToSignalMapping` in the role `callSignal`

- `ISignalTriggering` that refers to an `ISignal` that in turn refers to a `SystemSignal` that is referenced by the same `ClientServerToSignalMapping` in the role `returnSignal`

shall exist.

⌋*()*

[constr_5274] `ISignalTriggering`s that represent the `callSignal` and `returnSignal` of the same `ClientServerOperation` on a `PhysicalChannel` shall be referenced by the same `ClientServerToSignalMapping` ⌈If on an `EthernetPhysicalChannel` an `ISignalTriggering` that refers to an `ISignal` that in turn refers to a `SystemSignal` that is referenced by a specific `ClientServerToSignalMapping` in the role `callSignal` is defined, then another `ISignalTriggering` shall be aggregated by the same `EthernetPhysicalChannel` and that `ISignalTriggering` shall refer to an `ISignal` that in turn refers to a `SystemSignal` that is referenced by the same `ClientServerToSignalMapping` in the role `returnSignal`, and vice versa.

⌋*()*

[constr_5306] Restriction of `DltLogChannel.logChannelId` attribute value ⌈The `DltLogChannel`.logChannelId attribute value shall be composed of maximum four ASCII characters.

⌋*()*

[constr_5307] Existence of `DltLogChannel.logChannelId` ⌈For each `DltLogChannel`, the attribute `logChannelId` shall exist when the Base ECU Configuration is defined.

⌋*()*

[constr_5308] Existence of `DltLogChannel.nonVerboseMode` ⌈For each `Dlt-LogChannel`, the attribute `nonVerboseMode` shall exist when the Base ECU Configuration is defined.

⌋*()*

[constr_5309] Existence of `DltConfig.sessionIdSupport` ⌈For each `DltConfig`, the attribute `sessionIdSupport` shall exist when the Base ECU Configuration is defined.

⌋*()*

[constr_5310] Existence of `DltConfig.timestampSupport` ⌈For each `DltConfig`, the attribute `timestampSupport` shall exist when the Base ECU Configuration is defined.

⌋*()*

[constr_5311] Existence of `DltLogChannel.logTraceDefaultLogThreshold` ⌈For each `DltLogChannel`, the attribute `logTraceDefaultLogThreshold` shall exist when the Base ECU Configuration is defined.

⌋*()*

[constr_5312] Existence of `DltLogChannel.defaultTraceState` ⌈For each `DltLogChannel`, the attribute `defaultTraceState` shall exist when the Base ECU Configuration is defined.

⌋*()*

[constr_5313] Existence of `DltLogChannel.txPduTriggering` ⌈For each `Dlt-LogChannel`, the reference to `PduTriggering` in the role `txPduTriggering` shall exist when the Base ECU Configuration is defined.

⌋*()*

[constr_5314] `DltLogChannel txPduTriggering` and `rxPduTriggering` shall be on the same network ⌈The `PduTriggering`s that are referenced by a `Dlt-LogChannel` in the role `txPduTriggering` and `rxPduTriggering` shall be aggregated by the same `PhysicalChannel`.

⌋*()*

[constr_5315] `FlexrayArTpConnection`s within the same `FlexrayArTpChannel` not allowed to have the same address information ⌈`FlexrayArTpConnection`s that are aggregated by the same or reverse `FlexrayArTpChannel` are not allowed to reference the same pair of `FlexrayArTpNode`s at the time when the Ecu configuration of the COM stack is created.

⌋*()*

[constr_5319] TCP endpoint using `TLS_SERVER` role can only serve provided service instances ⌈An `ApplicationEndpoint` that refers to `TlsCryptoSer-`

viceMapping with category TLS_SERVER in the role tlsCryptoMapping is only allowed to be referenced by ProvidedServiceInstances in the role localUnicastAddress in case that the ProvidedServiceInstance does not have a remoteUnicastAddress defined.

⌋*()*

[constr_5320] TCP endpoint using `TLS_CLIENT` role can only serve consumed service instances ⌈An ApplicationEndpoint that refers to TlsCryptoServiceMapping with category TLS_CLIENT in the role tlsCryptoMapping is only allowed to be referenced by ConsumedServiceInstances in the role localUnicastAddress in case that the ConsumedServiceInstance does not have a remoteUnicastAddress defined.

⌋*()*

[constr_5321] Value range of `Pdu.length` ⌈The value of Pdu.length shall be in the range of 0..4294967295 Bytes.

⌋*()*

[constr_5322] Value range of `ISignalToIPduMapping.startPosition` ⌈The value of ISignalToIPduMapping.startPosition shall be in the range of 0..4294967295 Bits.

⌋*()*

[constr_5323] Value range of `ISignalToIPduMapping.updateIndicationBitPosition` ⌈The value of ISignalToIPduMapping.updateIndicationBitPosition shall be in the range of 0..4294967295 Bits.

⌋*()*

[constr_5326] Each local `SocketAddress` of an `EcuInstance` shall reference an `EthernetCommunicationConnector` in the role `connector` or `multicastConnector` ⌈If an EcuInstance uses a SocketAddress as local address, the SocketAddress shall refer to an EthernetCommunicationConnector of the EcuInstance, either via SocketAddress.connector if the SocketAddress represents a unicast address, or via SocketAddress.multicastConnector if the SocketAddress represents a multicast address.

⌋*()*

[constr_5327] Existence of attribute `CpSoftwareCluster.category` ⌈For each CpSoftwareCluster, attribute category shall exist at the time when the definition of the resource pool is finished.

⌋*()*

[constr_5328] Ecu Extract shall only contain outerPort `DataMapping`s ⌈The System with category ECU_EXTRACT shall only contain DataMappings for VariableDataPrototypes, ClientServerOperations or Triggers that are referenced in

the context of a `PortPrototype` of the `SwComponentType` that in turn is referenced by the `RootSwCompositionPrototype` at the time when the RTE is generated.

⌋*()*

[constr_5329] `SynchronousServerCallPoint`s for cross cluster communication are not supported ⌈A `ClientServerOperation` in the context of `PortPrototype` which is referenced by a `PortElementToCommunicationResourceMapping` in the role `clientServerOperation` is not allowed

- to be referenced by a `SynchronousServerCallPoint.operation` or

- to be connected to another `ClientServerOperation` in the context of a `PortPrototype` that in turn is referenced by `SynchronousServerCallPoint.operation`

at the time when the SW_CLUSTER_SYSTEM_DESCRIPTION is complete.

⌋*()*

[constr_5330] `ServiceInterface` elements shall belong to exactly one `ServiceInterface` ⌈If an element like

- a `VariableDataPrototype` that represents a `ServiceInterface event`

- a `ClientServerOperation` that represents a `ServiceInterface method`

- a `Collection` with `collectionSemantics` SO_SERVICE_FIELD that represents a `ServiceInterface field`

- a `Collection` with `collectionSemantics` SO_SERVICE_FIRE_AND_FORGET_METHOD that represents a "fire & forget" `method`

is referenced in the role `element` by a `Collection` that has the `collectionSemantics` set to SO_SERVICE_INTERFACE then this element shall not be referenced by any other `Collection` element that has the `collectionSemantics` SO_SERVICE_INTERFACE in the scope of the `System`.

⌋*()*

[constr_5331] No IP multicast in case of TCP ⌈The `ApplicationEndpoint` that is referenced in the role `eventMulticastAddress` from an `EventHandler` is only allowed to aggregate `UdpTp` in the role `tpConfiguration`.

⌋*()*

[constr_5334] Supported values for `CryptoServiceKey.length` ⌈The values defined for `CryptoServiceKey.length` shall be multiple of 8.

⌋*()*

[constr_5335] `CpSoftwareCluster.softwareClusterId` shall be unique in the scope of an `EcuInstance` ⌈The `softwareClusterId` shall be unique for each

CpSoftwareCluster that is mapped to the same EcuInstance with the CpSoftwareClusterToEcuInstanceMapping.

⌋*()*

[constr_5336] Existence of CpSoftwareCluster.softwareClusterId ⌈For each CpSoftwareCluster, attribute softwareClusterId shall exist at the time when when the SW_CLUSTER_SYSTEM_DESCRIPTION is complete.

⌋*()*

[constr_5337] All CpSoftwareClusterToEcuInstanceMappings that are referencing the same EcuInstance shall define the same machineId ⌈All CpSoftwareClusterToEcuInstanceMappings that define a machineId and are referencing the same EcuInstance in the role ecuInstance shall have the same CpSoftwareClusterToEcuInstanceMapping.machineId value set at the time when the System Description is finished.

⌋*()*

[constr_5344] Applicable `transferProperty` for GroupSignal and `ISignalGroup` ⌈

`transferProperty` on `ISignalGroup`	`transferProperty` on Group Signals	Semantic
• not set	Not set or `pending` for all Group Signals	Update of the Signal Group and update of Group Signals will not trigger transmission of the `ISignalIPdu`.
	Subset of Group Signals has `transferProperty` set to either `triggered` or `triggeredWithoutRepetition` and the other Group Signals have `transferProperty` either not set or set to `pending`.	Update of Signal Group marks the `ISignalIPdu` for transmission.
	Subset of Group Signals has `transferProperty` set to either `triggeredOnChange` or `triggeredOnChangeWithoutRepetition` and the other Group Signals have `transferProperty` either not set or set to `pending`.	Update of Signal Group and change of a Group Signal that has `transferProperty` set to `triggeredOnChange` or `triggeredOnChangeWithoutRepetition` causes immediate transmission of the `ISignalIPdu`.
• `pending`	Not set or `pending` for all Group Signals	Update of the Signal Group and update of Group Signals will not trigger transmission of the `ISignalIPdu`. See [TPS_SYST_02199].
• `triggered` • `triggeredWithoutRepetition`	Not set or `pending` for all Group Signals	Update of the Signal Group marks the `ISignalIPdu` for transmission. See [TPS_SYST_02199]
	Subset of Group Signals has `transferProperty` set to either `triggered` or `triggeredWithoutRepetition` and the other Group Signals have `transferProperty` either not set or set to `pending`.	Update of Signal Group marks the `ISignalIPdu` for transmission. See [TPS_SYST_02200].
• `triggeredOnChange` • `triggeredOnChangeWithoutRepetition`	Not set or `pending` for all Group Signals	Update of Signal Group causes immediate transmission of the `ISignalIPdu`. See [TPS_SYST_02199]
	Subset of Group Signals has transfer Property set to either `triggeredOnChange` or `triggeredOnChangeWithoutRepetition` and the other Group Signals have `transferProperty` either not set or set to `pending`.	Update of Signal Group and change of a Group Signal that has `transferProperty` set to `triggeredOnChange` or `triggeredOnChangeWithoutRepetition` causes immediate transmission of the `ISignalIPdu`. See [TPS_SYST_02200].

⌋*()*

[constr_5359] `CpSoftwareClusterBinaryManifestDescriptor.softwareClusterId` shall be identical to `CpSoftwareCluster.softwareClusterId` ⌈The `CpSoftwareClusterBinaryManifestDescriptor.softwareClusterId` shall be identical to `CpSoftwareCluster.softwareClusterId` in case that the `softwareClusterId` is set in the `CpSoftwareCluster` that is referenced via `CpSoftwareClusterBinaryManifestDescriptor.cpSoftwareCluster`.

⌋*()*

[constr_5360] Cross cluster communication involving `NvBlockSwComponent-Type` is not supported ⌈A `PortElementToCommunicationResourceMapping` that is referencing a `CpSoftwareClusterCommunicationResource` in the role `communicationResource` is not allowed to reference:

- a `VariableDataPrototype` in the role `variableDataPrototype` that is defined in the context of a `PortPrototype` of a `NvBlockSwComponentType` typed by a `NvDataInterface` or

- a `VariableDataPrototype` in the role `variableDataPrototype` which is connected to another `VariableDataPrototype` that is defined in the context of a `PortPrototype` of a `NvBlockSwComponentType` and typed by a `Nv-DataInterface` or

- a `ClientServerOperation` in the role `clientServerOperation` that is defined in the context of a `PortPrototype` of a `NvBlockSwComponentType` typed by a `ClientServerInterface` or

- a `ClientServerOperation` in the role `clientServerOperation` which is connected to another `ClientServerOperation` that is defined in the context of a `PortPrototype` of a `NvBlockSwComponentType` typed by a `ClientServerInterface`

at the time when the SW_CLUSTER_SYSTEM_DESCRIPTION is complete.

⌋*()*

[constr_5361] MACsec configuration is allowed only on switch ports ⌈Only a `CouplingElement` with `couplingType` set to `switch` is allowed to aggregate a `CouplingPort` that in turn aggregates the `MacSecProps` in the role `macSecProps`.

⌋*()*

2.9 TPS_TimingExtensions

[constr_4500] Restricted usage of functions ⌈The functions *TIMEX_occurs*, *TIMEX_hasOccurred*, *TIMEX_timeSinceLastOccurrence*, *TIMEX_angleSinceLastOccurrence*, and *TIMEX_modeActive* can only be used for occurrence expressions, which are applied to events of type `TDEventComplex`.

⌋*()*

[constr_4501] Application rule for the occurrence expression in `TDEventComplex` ⌈The occurrence expression shall be specified such that it describes an *event* rather than a state. As a consequence the occurrence expression shall ensure that a complex timing event *could* only occur at the occurrence time of one of the referenced `TimingDescriptionEvent`s.

⌋*()*

[constr_4502] Use references only as function operands ⌈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.

⌋*()*

[constr_4503] Restricted usage of `AutosarOperationArgumentInstance` for Content Filter ⌈If a content filter is defined for an atomic event then references to `AutosarOperationArgumentInstance`s are only allowed if the atomic event is of type `TDEventOperation`. 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).

⌋*()*

[constr_4504] Restricted usage of `AgeConstraint` ⌈An `AgeConstraint` shall only be defined for events of type `TimingDescriptionEvent` associated with the receipt and reading of data.

⌋*()*

[constr_4505] Specifying minimum and maximum number of occurrences ⌈The minimum and maximum number of occurrences shall be specified such that the following holds: $0 \leq$ `minNumberOfOccurrences` $\leq$ `maxNumberOfOccurrences`.

⌋*()*

[constr_4506] Specifying minimum inter-arrival time and pattern length ⌈The minimum inter-arrival time and pattern length shall be specified such that the following holds: $0 <$ `minimumInterArrivalTime` $\leq$ `patternLength`.

⌋*()*

[constr_4507] Specifying pattern length, pattern jitter and patter period ⌈The pattern length, pattern jitter and pattern period shall be specified such that the following holds: `patternLength` $+$ `patternJitter` $<$ `patternPeriod`.

⌋*()*

[constr_4508] `TDEventVfb` shall reference `PortPrototypeBlueprint` only in Blueprints ⌈An event type `TDEventVfb` only shall reference `PortPrototypeBlueprint` in blueprints.

⌋*()*

[constr_4509] **Only `VfbTiming` shall be a Blueprint** ⌈Only the `VfbTiming` is blueprintable.

⌋*()*

[constr_4510] Specifying references to `RunnableEntity` and `VariableAccess` ⌈A `RunnableEntity` and `VariableAccess` shall be referenced at the same time if and only if the value of `tdEventSwcInternalBehaviorType` is "runnableEntity VariableAccess". These two references are not mutual exclusive.

⌋*()*

[constr_4511] **Validity of referencing `RunnableEntity`** ⌈A `RunnableEntity` shall be referenced if and only if the value of `tdEventSwcInternalBehaviorType` is "runnableEntityActivated", "runnableEntityStarted", "runnableEntityTerminated", or "runnableEntityVariableAccess".

⌋*()*

[constr_4512] Validity of referencing `VariableAccess` ⌈A `VariableAccess` shall be referenced if and only if the value of `tdEventSwcInternalBehaviorType` is "runnableEntityVariableAccess".

⌋*()*

[constr_4513] **`SynchronizationTimingConstraint` shall reference at least two events** ⌈In the case, that the `SynchronizationTimingConstraint` is imposed on events then at least two (2) timing description events shall be referenced.

⌋*()*

[constr_4514] **`SynchronizationTimingConstraint` shall reference at least two event chains** ⌈In the case, that the `SynchronizationTimingConstraint` is imposed on event chains then at least two (2) timing description event chains shall be referenced.

⌋*()*

[constr_4515] Orthogonality of `stimulus` and `response` in a `TimingDescriptionEventChain` ⌈The reference `TimingDescriptionEventChain.stimulus` and the reference `TimingDescriptionEventChain.response` shall not reference the same `TimingDescriptionEvent.shortName`.

⌋*()*

[constr_4516] **Specifying event chain `segment`s** ⌈If a `TimingDescriptionEventChain` consists of further event chain `segment`s then at least one sequence of event chain `segment`s shall exist from the event chain's `stimulus` to the `response`.

⌋*()*

[constr_4517] **Referencing no further event chain `segment`s** ⌈If a `TimingDescriptionEventChain` is not subdivided in further event chain `segment`s, then the

reference playing the role of `segment` shall reference this `TimingDescription-EventChain`. In other words, an event chain without any event chain `segment`s shall reference itself.

⌋*()*

[constr_4518] Specifying `stimulus` event and `response` event of first and last event chain segment ⌈The `stimulus` event of the first event chain segment and the `response` event of the last event chain segment shall reference the `stimulus` and `response` of the parent event chain the event chain segments directly belong to.

⌋*()*

[constr_4519] Specifying `patternLength` ⌈The `patternLength` shall be specified such that the following holds: $0 \leq max(\texttt{offset}) \leq \texttt{patternLength}$.

⌋*()*

[constr_4520] Specifying attribute `synchronizationConstraintType` ⌈The attribute `synchronizationConstraintType` shall be specified if the `SynchronizationTimingConstraint` is imposed on events.

⌋*()*

[constr_4521] Specifying attribute `synchronizationConstraintType` ⌈The attribute `synchronizationConstraintType` shall be specified if the `SynchronizationTimingConstraint` is imposed on event chains.

⌋*()*

[constr_4522] `SynchronizationTimingConstraint` shall either reference events or event chains ⌈The `SynchronizationTimingConstraint` shall either reference timing description events or timing description event chains, but not both at the same time.

⌋*()*

[constr_4523] Specifying attributes `maxCycles` and `maxSlots` ⌈The optional attributes `maxCycles` and `maxSlots` shall never be specified in any element `EOCExecutableEntityRefGroup` that is part of a hierarchical execution order constraint.

⌋*()*

[constr_4524] Referencing `TimingDescriptionEvent` ⌈Any element `EOCExecutableEntityRefGroup` that is part of a hierarchical execution order constraint shall not reference any timing description event `TimingDescriptionEvent`.

⌋*()*

[constr_4525] Precedence of successor relationships `successor` and `directSuccessor` ⌈The successor relationships `successor` and `directSuccessor` take always precedence over the `ordered` multiplicity of the association `nestedElement`.

⌋*()*

[constr_4526] **Specifying `maxCycles` and `maxSlots` in a Repetitive Execution Order Constraint** ⌈The optional attributes `maxCycles` and `maxSlots` shall be specified only by the *root* group of executable entity references `EOCExecutableEntityRefGroup`.

⌋*()*

[constr_4527] **Referencing `TimingDescriptionEvent` in a Repetitive Execution Order Constraint** ⌈The `TimingDescriptionEvent` shall be specified only by the *root* group of executable entity references `EOCExecutableEntityRefGroup`.

⌋*()*

[constr_4528] **The *root* `EOCExecutableEntityRefGroup` shall reference only `EOCExecutableEntityRefGroup`s** ⌈The *root* `EOCExecutableEntityRefGroup` shall reference only groups of executable entity references respectively event references grouped by the element `EOCExecutableEntityRefGroup`s.

⌋*()*

[constr_4529] **Number of nested elements referenced by the *root* `EOCExecutableEntityRefGroup`** ⌈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** ⌈The `EOCExecutableEntityRefGroup` representing a cycle shall reference only executable entity references `EOCExecutableEntityRef`s respectively event references `EOCEventRef`s.

⌋*()*

[constr_4531] **Number of nested elements referenced by `EOCExecutableEntityRefGroup` representing a cycle** ⌈The number of nested elements referenced by a `EOCExecutableEntityRefGroup` representing a cycle shall be exactly the number given by the attribute `maxSlots`.

⌋*()*

[constr_4532] **Successor relationship is not self-referencing** ⌈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** ⌈The maximum number of successor relationships, namely `successor` or `directSuccessor`, between two `EOCExecutableEntityRef`s, between two `EOCEventRef`s, between two `EO-`

CExecutableEntityRefGroups, between an EOCExecutableEntityRef and an EOCExecutableEntityRefGroup, or between an EOCEventRef and an EOCExecutableEntityRefGroup is one (1).

⌋*()*

[constr_4534] Maximum number of `directSuccessor` relationships ⌈The number of directSuccessor relationships of an EOCExecutableEntityRef, an EOCEventRef, or an EOCExecutableEntityRefGroup shall not exceed the number of independent execution units available in a system.

⌋*()*

[constr_4536] Compatible recurrence of any `ExecutableEntity` ⌈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` ⌈An EOCExecutableEntityRef respectively EOCEventRef or an EOCExecutableEntityRefGroup shall reference only EOCExecutableEntityRefs, respectively all EOCEventRefs, or EOCExecutableEntityRefGroups which are part of the same ExecutionOrderConstraint.

⌋*()*

[constr_4538] Hierarchical Execution Order Constraint: `EOCExecutableEntityRef`, `EOCEventRef`, and `EOCExecutableEntityRefGroup` shall be target or source of a successor relationship ⌈In a given Hierarchical Execution Order Constraint, each EOCExecutableEntityRef, EOCEventRef, and EOCExecutableEntityRefGroup which is not part of an EOCExecutableEntityRefGroup shall be target or source of at least one successor relationship.

⌋*()*

[constr_4539] The successor relationships `successor` and `directSuccessor` shall not be used ⌈The successor relationships successor and directSuccessor shall not be used in a Repetitive Execution Order Constraint.

⌋*()*

[constr_4540] `maxCycles` and `maxSlots` shall not be zero ⌈If the optional attributes maxCycles and maxSlots are used, then the values of the optional attributes maxCycles and maxSlots shall be greater than zero (0).

⌋*()*

[constr_4541] `EOCExecutableEntityRef` **shall reference** `ExecutableEntity` **in Ordinary Execution Order Constraint** ⌈In an Ordinary Execution Order Constraint all `EOCExecutableEntityRef`s shall reference an `ExecutableEntity`.

⌋*()*

[constr_4542] `EOCExecutableEntityRef` **shall reference** `ExecutableEntity` **in Hierarchical Execution Order Constraint** ⌈In an Hierarchical Execution Order Constraint all `EOCExecutableEntityRef`s shall reference an `ExecutableEntity`.

⌋*()*

[constr_4543] Maximum value of the parameter `minimumInterArrivalTime` ⌈The value of the parameter `minimumInterArrivalTime` shall be less than or equal the value of the parameter `period`.

⌋*()*

[constr_4544] Specifying `patternLength`, `patternJitter` **and** `patternPeriod` ⌈The pattern length, pattern jitter and pattern period shall be specified such that the following holds: $patternLength + patternJitter < patternPeriod$.

⌋*()*

[constr_4545] Referring either `ExecutableEntity`**s or** `AbstractEvent`**s** ⌈An `ExecutionOrderConstraint` shall contain either only `EOCExecutableEntityRef` or only `EOCEventRef`, but not both. In the former case `ExecutableEntity`s are referenced and in the latter case `AbstractEvent`s are referenced.

⌋*()*

[constr_4546] Setting the attribute `isEvent` ⌈The value of the attribute `isEvent` shall be set to "TRUE" if and only if the execution order constraint refers to events only (refer to [constr_4545]). The value of the attribute `isEvent` shall be set to "FALSE" if and only if the execution order constraint refers to executable entities only (refer to [constr_4545]).

⌋*()*

[constr_4547]Setting the attribute `permitMultipleReferencesToEE` ⌈The value of the attribute `permitMultipleReferencesToEE` shall be specified if and only if the value of the attribute `isEvent` (refer to [constr_4546]) is set to "FALSE". In other words specifying whether an executable entity is permitted to be referenced more than once in an execution order constraint is only allowed in case of an execution order constraint referring to executable entities only.

⌋*()*

[constr_4548] `EOCEventRef` **shall reference** `AbstractEvent` **in Ordinary Execution Order Constraint** ⌈In an Ordinary Execution Order Constraint all `EOCEventRef`s shall reference an `AbstractEvent`.

⌋*()*

[constr_4549] **`EOCEventRef`** **shall reference** **`AbstractEvent`** **in Hierarchical Execution Order Constraint** ⌈In an Hierarchical Execution Order Constraint all `EOCEventRef`s shall reference an `AbstractEvent`.

⌋*()*

[constr_4550] **A Hierarchical Execution Order Constraint shall have an unambiguous root** **`EOCExecutableEntityRefGroup`** ⌈A Hierarchical Execution Order Constraint may contain multiple `orderedElement`s, which may be any combination of any number of `EOCExecutableEntityRef`s respectively `EOCEventRef`s and `EOCExecutableEntityRefGroup`s. Among these needs to be exactly one `EOCExecutableEntityRefGroup` being neither target nor source of any `successor` or `directSuccessor` relationship. This `EOCExecutableEntityRefGroup` is the *root* of the Hierarchical Execution Order Constraint.

⌋*()*

[constr_4551] **Use only Numericals in** **`TDEventOccurrenceExpression`** ⌈The target data prototype of the instance references of `variable` and `argument` shall be `Numerical`.

⌋*()*

[constr_4552] **Restricted usage of** **`AutosarVariableInstance`** **for Content Filter** ⌈If a content filter is defined for an atomic event then references to `AutosarVariableInstance`s 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_4553] **Usage of optional attribute** **`ignoreOrderAllowed`** ⌈This optional attribute shall only be used in the context of Logical Execution Time when an `EOCExecutableEntityRefGroup` is used to specify clusters of executable entities — executable entities cluster.

⌋*()*

[constr_4554] **Usage of optional directed association/reference** **`letInterval`** ⌈This optional directed association/reference shall only be used in the context of Logical Execution Time when an `EOCExecutableEntityRefGroup` is used to specify clusters of executable entities — executable entities cluster.

⌋*()*

[constr_4555] **Usage of the category value `LET_RELEASE` in** **`TimingDescriptionEvent`** ⌈The value `LET_RELEASE` 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, plays the role of a LET interval.

⌋*()*

[constr_4556] Usage of the category value `LET_TERMINATE` in `TimingDescriptionEvent` ⌈The value `LET_TERMINATE` 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, plays the role of a LET interval.

⌋*()*

[constr_4557] Usage of the category value `LET_INTERVAL` in `TimingDescriptionEventChain` ⌈The value `LET_INTERVAL` of the attribute `category` of a `TimingDescriptionEventChain` shall be set if and only if the timing description event chain references both:

- a timing description event playing the role stimulus and the value of the `category` of this referenced `TimingDescriptionEvent` is set to `LET_RELEASE` and

- a timing description event playing the role response and the value of the `category` of this referenced `TimingDescriptionEvent` is set to `LET_TERMINATE`

⌋*()*

[constr_4558] Applicability of LET semantics ⌈The LET semantics applies only to *implicit* sender-receiver communication.

⌋*()*

[constr_4559] Restriction of `TimingDescriptionEvent.category` ⌈Any `TimingDescriptionEvent.category` not in the list in [TPS_TIMEX_00056] shall be ignored.

⌋*()*

[constr_4560] Restriction of `TimingDescriptionEventChain.category` ⌈Any `TimingDescriptionEventChain.category` not in the list in [TPS_TIMEX_00110] shall be ignored.

⌋*()*

[constr_4561]{DRAFT} **Usage of the category value `DISPATCH_ENTRY_POINT` in `TimingDescriptionEvent`** ⌈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]{DRAFT} **Usage of the category value `DISPATCH_EXIT_POINT` in `TimingDescriptionEvent`** ⌈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.

⌋*()*

[constr_4563]{DRAFT} **`TDCpSoftwareClusterMapping` shall reference only dispatchers or LET intervals** ⌈The element `TDCpSoftwareClusterMapping` shall reference as timing description either `TDEventComplex` with the `category` set to `DISPATCH_ENTRY_POINT`, or `TimingDescriptionEventChain` with the `category` set to `LET_INTERVAL`.

⌋*()*

[constr_4564]{DRAFT} **`TDCpSoftwareClusterResourceMapping` shall reference only dispatchers or LET intervals** ⌈The element `TDCpSoftwareClusterResourceMapping` shall reference as timing description either `TDEventComplex` with the `category` set to `DISPATCH_ENTRY_POINT`, or `TimingDescriptionEventChain` with the `category` set to `LET_INTERVAL`.

⌋*()*

[constr_4565]{DRAFT} **Consistency of `TDCpSoftwareClusterMapping` and `TDCpSoftwareClusterResourceMapping`** ⌈The timing descriptions referenced by the element `TDCpSoftwareClusterMapping` and the element `TDCpSoftwareClusterResourceMapping` shall be consistent.

⌋*()*

[constr_4566]{DRAFT} **`SystemTiming` describing timing of software clusters and category of `System`** ⌈A `SystemTiming` used to describe the timing of one or more software clusters shall reference a `System` with the `category` set to `SW_CLUSTER_SYSTEM_DESCRIPTION` (see [TPS_SYST_01003])

⌋*()*

[constr_4567]{DRAFT} **Reference `provider` of `TDCpSoftwareClusterMapping`** ⌈The reference `provider` of `TDCpSoftwareClusterMapping` shall point to a `CpSoftwareCluster` which represents a host software cluster.

⌋*()*

[constr_4568]{DRAFT} **Reference `requestor` of `TDCpSoftwareClusterMapping`** ⌈The reference `requestor` of `TDCpSoftwareClusterMapping` shall point to a `CpSoftwareCluster` which represents an application software cluster.

⌋*()*

[constr_6816]{DRAFT} **Restricted usage of `TimingDescriptionEventChain.isPipeliningPermitted` in `TimingDescriptionEventChain`** ⌈The attribute `isPipeliningPermitted` shall only exist if the `TimingDescriptionEventChain`.category=SL_LET_INTERVAL.

⌋*()*

[constr_6817]{DRAFT} **Restricted usage of `TimingDescriptionEvent.clockReference`** ⌈The reference `TimingDescriptionEvent.clockReference` shall exist if (and only if), the `TimingDescriptionEvent` has a:

- `category`=SL_LET_RELEASE or
- `category`=SL_LET_TERMINATE

⌋*()*

[constr_6818] Existence of `EventTriggeringConstraint.event` ⌈For each `EventTriggeringConstraint`, the reference in the role `event` shall exist **at the imposition time associated with the concrete subclass of TimingDescriptionEvent**.

⌋*()*

[constr_6819] Existence of `PeriodicEventTriggering.jitter` ⌈For each `PeriodicEventTriggering`, the attribute `jitter` shall exist **at the imposition time associated with the concrete subclass of TimingDescriptionEvent**.

⌋*()*

[constr_6820] Existence of `PeriodicEventTriggering.minimumInterArrivalTime` ⌈For each `PeriodicEventTriggering`, the attribute `minimumInterArrivalTime` shall exist **at the imposition time associated with the concrete subclass of TimingDescriptionEvent**.

⌋*()*

[constr_6821] Existence of `PeriodicEventTriggering.period` ⌈For each `PeriodicEventTriggering`, the attribute `period` shall exist **at the imposition time associated with the concrete subclass of TimingDescriptionEvent**.

⌋*()*

[constr_6822] Existence of `SporadicEventTriggering.maximumInterArrivalTime` ⌈For each `SporadicEventTriggering`, the attribute `maximumInterArrivalTime` shall exist **at the imposition time associated with the concrete subclass of TimingDescriptionEvent**.

⌋*()*

[constr_6823] **Existence of `SporadicEventTriggering.minimumInterArrivalTime`** ⌈For each `SporadicEventTriggering`, the attribute `minimumInterArrivalTime` shall exist **at the imposition time associated with the concrete subclass of TimingDescriptionEvent**.

⌋*()*

[constr_6824] **Existence of `ConcretePatternEventTriggering.patternLength`** ⌈For each `ConcretePatternEventTriggering`, the attribute `patternLength` shall exist **at the imposition time associated with the concrete subclass of TimingDescriptionEvent**.

⌋*()*

[constr_6825] Existence of `BurstPatternEventTriggering.maxNumberOfOccurrences` ⌈For each `BurstPatternEventTriggering`, the attribute `maxNumberOfOccurrences` shall exist **at the imposition time associated with the concrete subclass of TimingDescriptionEvent**.

⌋*()*

[constr_6826] Existence of `BurstPatternEventTriggering.minimumInterArrivalTime` ⌈For each `BurstPatternEventTriggering`, the attribute `minimumInterArrivalTime` shall exist **at the imposition time associated with the concrete subclass of TimingDescriptionEvent**.

⌋*()*

[constr_6827] Existence of `BurstPatternEventTriggering.patternLength` ⌈For each `BurstPatternEventTriggering`, the attribute `patternLength` shall exist **at the imposition time associated with the concrete subclass of TimingDescriptionEvent**.

⌋*()*

[constr_6828] Existence of `ArbitraryEventTriggering.minimumDistance` ⌈For each `ArbitraryEventTriggering`, the reference in the role `minimumDistance` shall exist at least once **at the imposition time associated with the concrete subclass of TimingDescriptionEvent**.

⌋*()*

[constr_6829] Existence of `ArbitraryEventTriggering.maximumDistance` ⌈For each `ArbitraryEventTriggering`, the reference in the role `maximumDistance` shall exist at least once **at the imposition time associated with the concrete subclass of TimingDescriptionEvent**.

⌋*()*

[constr_6830] **Existence of `ConfidenceInterval.lowerBound`** ⌈For each `ConfidenceInterval`, the attribute `lowerBound` shall exist **at the imposition**

time associated with the concrete subclass of TimingDescriptionEvent.

⌋*()*

[constr_6831] Existence of ConfidenceInterval.propability ⌈For each ConfidenceInterval, the attribute propability shall exist **at the imposition time associated with the concrete subclass of TimingDescriptionEvent**.

⌋*()*

[constr_6832] Existence of ConfidenceInterval.upperBound ⌈For each ConfidenceInterval, the attribute upperBound shall exist **at the imposition time associated with the concrete subclass of TimingDescriptionEvent**.

⌋*()*

[constr_6833] Existence of ExecutionOrderConstraint.orderedElement ⌈For each ExecutionOrderConstraint, the attribute orderedElement shall exist at least once **at the imposition time associated with the concrete subclass of TimingDescriptionEvent**.

⌋*()*

[constr_6834] Existence of EOCExecutableEntityRefGroup.nestedElement ⌈For each EOCExecutableEntityRefGroup, the reference in the role nestedElement shall exist at least once **at the imposition time associated with the concrete subclass of TimingDescriptionEvent**.

⌋*()*

[constr_6835] Existence of ExecutionTimeConstraint.executionTimeType ⌈For each ExecutionTimeConstraint, the attribute executionTimeType shall exist **at the time when the Swc Timing Description is complete**.

⌋*()*

[constr_6836] Existence of ExecutionTimeConstraint.executable ⌈For each ExecutionTimeConstraint, the reference to ExecutableEntity in the role executable shall exist **at the time when the Swc Timing Description is complete**.

⌋*()*

[constr_6837] Existence of LatencyTimingConstraint.latencyConstraintType ⌈For each LatencyTimingConstraint, the attribute latencyConstraintType shall exist **at the imposition time associated with the concrete subclass of TimingDescriptionEvent**.

⌋*()*

[constr_6838] Existence of `LatencyTimingConstraint.maximum` ⌈For each `LatencyTimingConstraint`, the attribute `maximum` shall exist **at the imposition time associated with the concrete subclass of TimingDescriptionEvent**.

⌋*()*

[constr_6839] Existence of `LatencyTimingConstraint.minimum` ⌈For each `LatencyTimingConstraint`, the attribute `minimum` shall exist **at the imposition time associated with the concrete subclass of TimingDescriptionEvent**.

⌋*()*

[constr_6840] Existence of `LatencyTimingConstraint.nominal` ⌈For each `LatencyTimingConstraint`, the attribute `nominal` shall exist **at the imposition time associated with the concrete subclass of TimingDescriptionEvent**.

⌋*()*

[constr_6841] Existence of `LatencyTimingConstraint.scope` ⌈For each `LatencyTimingConstraint`, the reference in the role `scope` shall exist at least once **at the imposition time associated with the concrete subclass of TimingDescriptionEvent**.

⌋*()*

[constr_6842] Existence of `OffsetTimingConstraint.maximum` ⌈For each `OffsetTimingConstraint`, the attribute `maximum` shall exist **at the imposition time associated with the concrete subclass of TimingDescriptionEvent**.

⌋*()*

[constr_6843] Existence of `OffsetTimingConstraint.minimum` ⌈For each `OffsetTimingConstraint`, the attribute `minimum` shall exist **at the imposition time associated with the concrete subclass of TimingDescriptionEvent**.

⌋*()*

[constr_6844] Existence of `OffsetTimingConstraint.source` ⌈For each `OffsetTimingConstraint`, the reference in the role `source` shall exist at least once **at the imposition time associated with the concrete subclass of TimingDescriptionEvent**.

⌋*()*

[constr_6845] Existence of `OffsetTimingConstraint.target` ⌈For each `OffsetTimingConstraint`, the reference in the role `target` shall exist

at least once **at the imposition time associated with the concrete subclass of TimingDescriptionEvent**.

⌋*()*

[constr_6846] **Existence of `SynchronizationTimingConstraint.synchronizationConstraintType`** ⌈For each `SynchronizationTimingConstraint`, the attribute `synchronizationConstraintType` shall exist **at the imposition time associated with the concrete subclass of TimingDescriptionEvent**.

⌋*()*

[constr_6847] **Existence of `SynchronizationTimingConstraint.tolerance`** ⌈For each `SynchronizationTimingConstraint`, the attribute `tolerance` shall exist **at the imposition time associated with the concrete subclass of TimingDescriptionEvent**.

⌋*()*

[constr_6848] Existence of `VfbTiming.component` ⌈For each `VfbTiming`, the reference to `SwComponentType` in the role `component` shall exist **at the time when the VFB Timing Description is complete**.

⌋*()*

[constr_6849] Existence of `SystemTiming.system` ⌈For each `SystemTiming`, the reference to `System` in the role `system` shall exist **at the time when the System Timing Description is complete**.

⌋*()*

[constr_6850] Existence of `BswModuleTiming.behavior` ⌈For each `BswModuleTiming`, the reference to `BswInternalBehavior` in the role `behavior` shall exist **at the time when the Bsw Timing Description is complete**.

⌋*()*

[constr_6851] Existence of `BswCompositionTiming.implementation` ⌈For each `BswCompositionTiming`, the reference to `BswImplementation` in the role `implementation` shall exist at least once **at the time when the Bsw Timing Description is complete**.

⌋*()*

[constr_6852] Existence of `EcuTiming.ecuConfiguration` ⌈For each `EcuTiming`, the reference to `EcucValueCollection` in the role `ecuConfiguration` shall exist at least once **at the time when the Ecu Timing Description is complete**.

⌋*()*

[constr_6853] **Existence of `ModeInBswInstanceRef.contextModeDeclarationGroupPrototype`** ⌈For each `ModeInBswInstanceRef`, the reference to `ModeDeclarationGroupPrototype` in the role `contextModeDeclarationGroupPrototype` shall exist at least once **at the time when the Bsw Timing Description is complete**.

⌋*()*

[constr_6854] Existence of `ModeInBswInstanceRef.targetModeDeclaration` ⌈For each `ModeInBswInstanceRef`, the reference to `ModeDeclaration` in the role `targetModeDeclaration` shall exist at least once **at the time when the Bsw Timing Description is complete**.

⌋*()*

[constr_6855] **Existence of `ModeInSwcInstanceRef.contextModeDeclarationGroupPrototype`** ⌈For each `ModeInSwcInstanceRef`, the reference to `ModeDeclarationGroupPrototype` in the role `contextModeDeclarationGroupPrototype` shall exist at least once **at the time when the Swc Timing Description is complete**.

⌋*()*

[constr_6856] **Existence of `ModeInSwcInstanceRef.contextPort`** ⌈For each `ModeInSwcInstanceRef`, the reference to `PortPrototype` in the role `contextPort` shall exist at least once **at the time when the Swc Timing Description is complete**.

⌋*()*

[constr_6857] Existence of `ModeInSwcInstanceRef.targetModeDeclaration` ⌈For each `ModeInSwcInstanceRef`, the reference to `ModeDeclaration` in the role `targetModeDeclaration` shall exist at least once **at the time when the Swc Timing Description is complete**.

⌋*()*

[constr_6858] **Existence of `TDEventBswInternalBehavior.tdEventBswInternalBehaviorType`** ⌈For each `TDEventBswInternalBehavior`, the attribute `tdEventBswInternalBehaviorType` shall exist **at the time when the Bsw Timing Description is complete**.

⌋*()*

[constr_6859] Existence of `TDEventBswInternalBehavior.bswModuleEntity` ⌈For each `TDEventBswInternalBehavior`, the reference to `BswModuleEntity` in the role `bswModuleEntity` shall exist **at the time when the Bsw Timing Description is complete**.

⌋*()*

[constr_6860] Existence of `TDEventBswModule.tdEventBswModuleType` ⌈For each `TDEventBswModule`, the attribute `tdEventBswModuleType` shall exist **at the time when the Bsw Timing Description is complete**.

⌋*()*

[constr_6861] Existence of `TDEventBswModule.bswModuleEntry` ⌈For each `TDEventBswModule`, the reference to `BswModuleEntry` in the role `bswModuleEntry` shall exist **at the time when the Bsw Timing Description is complete**.

⌋*()*

[constr_6862] Existence of `TDEventBswModeDeclaration.tdEventBswModeDeclarationType` ⌈For each `TDEventBswModeDeclaration`, the attribute `tdEventBswModeDeclarationType` shall exist **at the time when the Bsw Timing Description is complete**.

⌋*()*

[constr_6863] Existence of `TDEventBswModeDeclaration.modeDeclaration` ⌈For each `TDEventBswModeDeclaration`, the reference to `ModeDeclarationGroupPrototype` in the role `modeDeclaration` shall exist **at the time when the Bsw Timing Description is complete**.

⌋*()*

[constr_6864] Existence of `TDEventISignal.tdEventType` ⌈For each `TDEventISignal`, the attribute `tdEventType` shall exist **at the time when the Ecu Timing Description is complete**.

⌋*()*

[constr_6865] Existence of `TDEventISignal.iSignal` ⌈For each `TDEventISignal`, the reference to `ISignal` in the role `iSignal` shall exist **at the time when the Ecu Timing Description is complete**.

⌋*()*

[constr_6866] Existence of `TDEventISignal.physicalChannel` ⌈For each `TDEventISignal`, the reference to `PhysicalChannel` in the role `physicalChannel` shall exist **at the time when the Ecu Timing Description is complete**.

⌋*()*

[constr_6867] Existence of `TDEventIPdu.tdEventType` ⌈For each `TDEventIPdu`, the attribute `tdEventType` shall exist **at the time when the Ecu Timing Description is complete**.

⌋*()*

[constr_6868] Existence of `TDEventIPdu.iPdu` ⌈For each `TDEventIPdu`, the reference to `IPdu` in the role `iPdu` shall exist **at the time when the Ecu Timing Description is complete**.

⌋*()*

[constr_6869] Existence of `TDEventIPdu.physicalChannel` ⌈For each `TDEventIPdu`, the reference to `PhysicalChannel` in the role `physicalChannel` shall exist **at the time when the Ecu Timing Description is complete**.

⌋*()*

[constr_6870] Existence of `TDEventFrame.tdEventType` ⌈For each `TDEventFrame`, the attribute `tdEventType` shall exist **at the time when the Ecu Timing Description is complete**.

⌋*()*

[constr_6871] Existence of `TDEventFrame.frame` ⌈For each `TDEventFrame`, the reference to `Frame` in the role `frame` shall exist **at the time when the Ecu Timing Description is complete**.

⌋*()*

[constr_6872] Existence of `TDEventFrame.physicalChannel` ⌈For each `TDEventFrame`, the reference to `PhysicalChannel` in the role `physicalChannel` shall exist **at the time when the Ecu Timing Description is complete**.

⌋*()*

[constr_6873] Existence of `TDEventFrameEthernet.tdEventType` ⌈For each `TDEventFrameEthernet`, the attribute `tdEventType` shall exist **at the time when the Ecu Timing Description is complete**.

⌋*()*

[constr_6874] Existence of `TDHeaderIdRange.maxHeaderId` ⌈For each `TDHeaderIdRange`, the attribute `maxHeaderId` shall exist **at the time when the Ecu Timing Description is complete**.

⌋*()*

[constr_6875] Existence of `TDHeaderIdRange.minHeaderId` ⌈For each `TDHeaderIdRange`, the attribute `minHeaderId` shall exist **at the time when the Ecu Timing Description is complete**.

⌋*()*

[constr_6876] Existence of `TDEventCycleStart.cycleRepetition` ⌈For each `TDEventCycleStart`, the attribute `cycleRepetition` shall exist **at the time when the Ecu Timing Description is complete**.

⌋*()*

[constr_6877] **Existence of** **TDEventFrClusterCycleStart.frCluster** ⌈For each TDEventFrClusterCycleStart, the attribute frCluster shall exist **at the time when the Ecu Timing Description is complete**.

⌋*()*

[constr_6878] **Existence of** **TDEventTTCanCycleStart.ttCanCluster** ⌈For each TDEventTTCanCycleStart, the attribute ttCanCluster shall exist **at the time when the Ecu Timing Description is complete**.

⌋*()*

[constr_6879] **Existence of** **TDEventOccurrenceExpression.formula** ⌈For each TDEventOccurrenceExpression, the attribute formula shall exist **at the imposition time associated with the concrete subclass of TimingDescriptionEvent**.

⌋*()*

[constr_6880] **Existence of** **AutosarVariableInstance.variableInstance** ⌈For each AutosarVariableInstance, the reference in the role variable-Instance shall exist **at the imposition time associated with the concrete subclass of TimingDescriptionEvent**.

⌋*()*

[constr_6881] **Existence of** **AutosarOperationArgumentInstance.operationArgumentInstance** ⌈For each AutosarOperationArgumentInstance, the reference in the role operationArgumentInstance shall exist **at the imposition time associated with the concrete subclass of TimingDescriptionEvent**.

⌋*()*

[constr_6882] **Existence of** **TDEventSwcInternalBehavior.tdEventSwcInternalBehaviorType** ⌈For each TDEventSwcInternalBehavior, the attribute tdEventSwcInternalBehaviorType shall exist **at the time when the Swc Timing Description is complete**.

⌋*()*

[constr_6883] **Existence of** **TDEventSwcInternalBehavior.runnable** ⌈For each TDEventSwcInternalBehavior, the reference to RunnableEntity in the role runnable shall exist **at the time when the Swc Timing Description is complete**.

⌋*()*

[constr_6884] **Existence of** **TDEventSwcInternalBehaviorReference.referencedTDEventSwc** ⌈For each TDEventSwcInternalBehaviorReference, the

reference to `TDEventSwc` in the role `referencedTDEventSwc` shall exist **at the time when the Swc Timing Description is complete**.

⌋*()*

[constr_6885] Existence of `TDEventVfbPort.isExternal` ⌈For each `TDEventVfbPort`, the attribute `isExternal` shall exist **at the time when the VFB Timing Description is complete**.

⌋*()*

[constr_6886] Existence of `TDEventVfbReference.referencedTDEventVfb` ⌈For each `TDEventVfbReference`, the reference to `TDEventVfb` in the role `referencedTDEventVfb` shall exist **at the time when the VFB Timing Description is complete**.

⌋*()*

[constr_6887] Existence of `TDEventVariableDataPrototype.tdEventVariableDataPrototypeType` ⌈For each `TDEventVariableDataPrototype`, the attribute `tdEventVariableDataPrototypeType` shall exist **at the time when the VFB Timing Description is complete**.

⌋*()*

[constr_6888] Existence of `TDEventVariableDataPrototype.dataElement` ⌈For each `TDEventVariableDataPrototype`, the reference to `VariableDataPrototype` in the role `dataElement` shall exist **at the time when the VFB Timing Description is complete**.

⌋*()*

[constr_6889] Existence of `TDEventOperation.tdEventOperationType` ⌈For each `TDEventOperation`, the attribute `tdEventOperationType` shall exist **at the time when the VFB Timing Description is complete**.

⌋*()*

[constr_6890] Existence of `TDEventOperation.operation` ⌈For each `TDEventOperation`, the reference to `ClientServerOperation` in the role `operation` shall exist **at the time when the VFB Timing Description is complete**.

⌋*()*

[constr_6891] Existence of `TDEventModeDeclaration.tdEventModeDeclarationType` ⌈For each `TDEventModeDeclaration`, the attribute `tdEventModeDeclarationType` shall exist **at the time when the VFB Timing Description is complete**.

⌋*()*

[constr_6892] Existence of `TDEventModeDeclaration.modeDeclaration` ⌈For each `TDEventModeDeclaration`, the reference to `ModeDeclarationGroupPro-`

totype in the role modeDeclaration shall exist **at the time when the VFB Timing Description is complete**.

⌋*()*

[constr_6893] Existence of TDEventTrigger.tdEventTriggerType ⌈For each TDEventTrigger, the attribute tdEventTriggerType shall exist **at the time when the VFB Timing Description is complete**.

⌋*()*

[constr_6894] Existence of TDEventTrigger.trigger ⌈For each TDEventTrigger, the reference to Trigger in the role trigger shall exist **at the time when the VFB Timing Description is complete**.

⌋*()*

[constr_6895] Existence of TimingDescriptionEventChain.response ⌈For each TimingDescriptionEventChain, the reference in the role response shall exist **at the imposition time associated with the concrete subclass of TimingDescriptionEvent**.

⌋*()*

[constr_6896] Existence of TimingDescriptionEventChain.stimulus ⌈For each TimingDescriptionEventChain, the reference in the role stimulus shall exist **at the imposition time associated with the concrete subclass of TimingDescriptionEvent**.

⌋*()*

[constr_6897] Existence of TimingDescriptionEventChain.segment ⌈For each TimingDescriptionEventChain, the reference in the role segment shall exist at least once **at the imposition time associated with the concrete subclass of TimingDescriptionEvent**.

⌋*()*

[constr_6898] Existence of ConcretePatternEventTriggering.offset ⌈For each ConcretePatternEventTriggering, the attribute offset shall exist **at the imposition time associated with the concrete subclass of TimingDescriptionEvent**.

⌋*()*

[constr_6899] Existence of ModeInSwcInstanceRef.base ⌈For each ModeInSwcInstanceRef, the reference to SwComponentType in the role base shall exist at least once **at the time when the Swc Timing Description is complete**.

⌋*()*

A   Mentioned Class Tables

Class	**ARElement** (abstract)
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::ARPackage
Note	An element that can be defined stand-alone, i.e. without being part of another element (except for packages of course).
Base	*ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*
Subclasses	AclObjectSet, AclOperation, AclPermission, AclRole, AliasNameSet, ApplicationPartition, *AutosarData Type*, *BaseType*, BlueprintMappingSet, BswEntryRelationshipSet, BswModuleDescription, BswModule Entry, BuildActionManifest, CalibrationParameterValueSet, ClientIdDefinitionSet, ClientServerInterfaceTo BswModuleEntryBlueprintMapping, Collection, CompuMethod, ConsistencyNeedsBlueprintSet, Constant Specification, ConstantSpecificationMappingSet, CpSoftwareCluster, CpSoftwareClusterBinaryManifest Descriptor, CpSoftwareClusterMappingSet, CpSoftwareClusterResourcePool, CryptoEllipticCurveProps, CryptoServiceCertificate, CryptoServiceKey, CryptoServicePrimitive, CryptoServiceQueue, Crypto SignatureScheme, DataConstr, DataExchangePoint, DataTransformationSet, DataTypeMappingSet, *DiagnosticCommonElement*, DiagnosticConnection, DiagnosticContributionSet, DltContext, DltEcu, Documentation, E2EProfileCompatibilityProps, EcucDefinitionCollection, EcucDestinationUriDefSet, EcucModuleConfigurationValues, EcucModuleDef, EcucValueCollection, EndToEndProtectionSet, EthIp Props, EthTcpIpIcmpProps, EthTcpIpProps, EvaluatedVariantSet, FMFeature, FMFeatureMap, FM FeatureModel, FMFeatureSelectionSet, FlatMap, GeneralPurposeConnection, HwCategory, HwElement, HwType, IPSecConfigProps, IPv6ExtHeaderFilterSet, *IdsCommonElement*, IdsDesign, *Implementation*, InterpolationRoutineMappingSet, J1939ControllerApplication, KeywordSet, LifeCycleInfoSet, LifeCycle StateDefinitionGroup, LogAndTraceMessageCollectionSet, MacSecGlobalKayProps, MacSecParticipant Set, McFunction, McGroup, ModeDeclarationGroup, ModeDeclarationMappingSet, OsTaskProxy, PhysicalDimension, PhysicalDimensionMappingSet, *PortInterface*, PortInterfaceMappingSet, Port PrototypeBlueprint, PostBuildVariantCriterion, PostBuildVariantCriterionValueSet, PredefinedVariant, RapidPrototypingScenario, SdgDef, SignalServiceTranslationPropsSet, SomeipSdClientEventGroup TimingConfig, SomeipSdClientServiceInstanceConfig, SomeipSdServerEventGroupTimingConfig, SomeipSdServerServiceInstanceConfig, SwAddrMethod, SwAxisType, SwComponentMapping Constraints, *SwComponentType*, SwRecordLayout, SwSystemconst, SwSystemconstantValueSet, Swc BswMapping, System, SystemSignal, SystemSignalGroup, TDCpSoftwareClusterMappingSet, Tcp OptionFilterSet, *TimingExtension*, TlsConnectionGroup, TlvDataIdDefinitionSet, TransformationProps Set, Unit, UnitGroup, ViewMapSet
Aggregated by	ARPackage.element

Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.1: ARElement

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	*ARObject*, *AtpBlueprint*, *AtpBlueprintable*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*
Aggregated by	ARPackage.arPackage, AUTOSAR.arPackage

Attribute	Type	Mult.	Kind	Note

▽

△

Class	ARPackage			
arPackage	ARPackage	*	aggr	This represents a sub package within an ARPackage, thus allowing for an unlimited package hierarchy. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=arPackage.shortName, arPackage.variation Point.shortLabel vh.latestBindingTime=blueprintDerivationTime xml.sequenceOffset=30
element	PackageableElement	*	aggr	Elements that are part of this package **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=element.shortName, element.variation Point.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=20
referenceBase	ReferenceBase	*	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:** atpSplitable **Tags:** atp.Splitkey=referenceBase.shortLabel xml.sequenceOffset=10

Table A.2: ARPackage

Class	AUTOSAR			
Package	M2::AUTOSARTemplates::AutosarTopLevelStructure			
Note	Root element of an AUTOSAR description, also the root element in corresponding XML documents. **Tags:**xml.globalElement=true			
Base	ARObject			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
adminData	AdminData	0..1	aggr	This represents the administrative data of an Autosar file. **Tags:**xml.sequenceOffset=10
arPackage	ARPackage	*	aggr	This is the top level package in an AUTOSAR model. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=arPackage.shortName, arPackage.variation Point.shortLabel vh.latestBindingTime=blueprintDerivationTime xml.sequenceOffset=30
fileInfo Comment	FileInfoComment	0..1	aggr	This represents a possibility to provide a structured comment in an AUTOSAR file. **Stereotypes:** atpStructuredComment **Tags:** xml.roleElement=true xml.sequenceOffset=-10 xml.typeElement=false
introduction	DocumentationBlock	0..1	aggr	This represents an introduction on the Autosar file. It is intended for example to represent disclaimers and legal notes. **Tags:**xml.sequenceOffset=20

Table A.3: AUTOSAR

Class	AbstractAccessPoint (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::AccessCount			
Note	Abstract class indicating an access point from an ExecutableEntity.			
Base	ARObject, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable, MultilanguageReferrable, Referrable			
Subclasses	AsynchronousServerCallResultPoint, ExternalTriggeringPointIdent, InternalTriggeringPoint, ModeAccess PointIdent, ModeSwitchPoint, ParameterAccess, ServerCallPoint, VariableAccess			
Aggregated by	AtpClassifier.atpFeature			
Attribute	Type	Mult.	Kind	Note
returnValue Provision	RteApiReturnValue ProvisionEnum	0..1	attr	This attribute controls the provision of return values for RTE APIs that correspond to the enclosing access point.

Table A.4: AbstractAccessPoint

Class	AbstractCanCommunicationControllerAttributes (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	For the configuration of the CanController parameters two different approaches can be used: 1. Providing exact values which are taken by the ECU developer (CanControllerConfiguration). 2. Providing ranges of values which are taken as requirements and have to be respected by the ECU developer (CanControllerConfigurationRequirements).			
Base	ARObject			
Subclasses	CanControllerConfiguration, CanControllerConfigurationRequirements			
Aggregated by	AbstractCanCommunicationController.canControllerAttributes			
Attribute	Type	Mult.	Kind	Note
canControllerFd Attributes	CanControllerFd Configuration	0..1	aggr	Bit timing related configuration of a CAN controller for payload and CRC of a CanFD frame. If this element exists the controller supports CanFD frames and the ECU developer shall take these values for the configuration of the CanFD controller.
canControllerFd Requirements	CanControllerFd Configuration Requirements	0..1	aggr	Additional CanFD ranges of the bit timing related configuration of a CanFD controller. If this element exists the controller supports CanFD frames and the ECU developer shall take these ranges as requirements for the configuration of the CanFD controller.
canControllerXl Attributes	CanControllerXl Configuration	0..1	aggr	Bit timing related configuration of a CAN controller for payload and CRC of a CanXL frame. If this element exists the controller supports CanXL frames and the ECU developer shall take these values for the configuration of the CanXL controller.
canControllerXl Requirements	CanControllerXl Configuration Requirements	0..1	aggr	Additional CanXL ranges of the bit timing related configuration of a CanXL controller. If this element exists the controller supports CanXL frames and the ECU developer shall take these ranges as requirements for the configuration of the CanXL controller.

Table A.5: AbstractCanCommunicationControllerAttributes

Class	AbstractClassTailoring
Package	M2::AUTOSARTemplates::CommonStructure::StandardizationTemplate::DataExchangePoint::Data FormatTailoring
Note	Tailoring of abstract classes in the AUTOSAR meta-model
Base	ARObject, ClassTailoring, DataFormatElementReference, Identifiable, MultilanguageReferrable, Referrable, SpecElementReference

▽

△

Class	AbstractClassTailoring			
Aggregated by	AggregationTailoring.typeTailoring, DataFormatTailoring.classTailoring, ReferenceTailoring.typeTailoring			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.6: AbstractClassTailoring

Class	<<atpMixedString>> *AbstractEnumerationValueVariationPoint* (abstract)			
Package	M2::AUTOSARTemplates::GenericStructure::VariantHandling::AttributeValueVariationPoints			
Note	This is an abstract EnumerationValueVariationPoint. It is introduced to support the case that additional attributes are required for particular purposes.			
Base	*ARObject*, *AttributeValueVariationPoint*, *FormulaExpression*, *SwSystemconstDependentFormula*			
Subclasses				
Aggregated by	VariationPointProxy.valueAccess			
Attribute	Type	Mult.	Kind	Note
base	Identifier	0..1	attr	This attribute reflects the base to be used in context of EnumerationMappingTable for this reference. **Tags:**xml.attribute=true
enumTable	Ref	0..1	attr	This represents the assigned enumeration table. **Tags:**xml.attribute=true

Table A.7: AbstractEnumerationValueVariationPoint

Class	*AbstractEthernetFrame* (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetFrame			
Note	Ethernet specific attributes to the Frame.			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Frame*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	GenericEthernetFrame, Ieee1722TpEthernetFrame, UserDefinedEthernetFrame			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.8: AbstractEthernetFrame

Class	*AbstractEvent* (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*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	*BswEvent*, *RTEEvent*			
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 A.9: AbstractEvent

Class	**AbstractImplementationDataType** (abstract)				
Package	M2::AUTOSARTemplates::CommonStructure::ImplementationDataTypes				
Note	This meta-class represents an abstract base class for different flavors of ImplementationDataType.				
Base	*ARElement*, ARObject, *AtpBlueprint*, AtpBlueprintable, *AtpClassifier*, *AtpType*, *AutosarDataType*, CollectableElement, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*				
Subclasses	ImplementationDataType				
Aggregated by	ARPackage.element				
Attribute	Type	Mult.	Kind	Note	
–	–	–	–	–	

Table A.10: AbstractImplementationDataType

Class	**AbstractImplementationDataTypeElement** (abstract)				
Package	M2::AUTOSARTemplates::CommonStructure::ImplementationDataTypes				
Note	This meta-class represents the ability to act as an abstract base class for specific derived meta-classes that support the modeling of ImplementationDataTypes for a particular language binding.				
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*				
Subclasses	ImplementationDataTypeElement				
Aggregated by	*AtpClassifier*.atpFeature				
Attribute	Type	Mult.	Kind	Note	
–	–	–	–	–	

Table A.11: AbstractImplementationDataTypeElement

Class	**AbstractMultiplicityRestriction** (abstract)			
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::ModelRestrictionTypes			
Note	Restriction that specifies the valid number of occurrences of an element in the current context.			
Base	*ARObject*			
Subclasses	*AttributeCondition*, MultiplicityRestrictionWithSeverity, *SdgAttribute*			
Attribute	Type	Mult.	Kind	Note
lowerMultiplicity	PositiveInteger	0..1	attr	Specifies the minimal number of times an object shall occur. If this primitive attribute is not set, then the object is optional.
upperMultiplicity	PositiveInteger	0..1	attr	Specifies the maximum number of times an object may occur. If this primitive attribute is not set, then there is no limit with respect to the maximum occurrence.
upperMultiplicityInfinite	Boolean	0..1	attr	This explicitly specifies, that the upper multiplicity is NOT restricted. Note: The use of 'upperMultiplicityInfinite' and 'upperMultiplicity' is mutual exclusive.

Table A.12: AbstractMultiplicityRestriction

Class	**AbstractProvidedPortPrototype** (abstract)
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components
Note	This abstract class provides the ability to become a provided PortPrototype.
Base	*ARObject*, *AtpBlueprintable*, *AtpFeature*, *AtpPrototype*, *Identifiable*, *MultilanguageReferrable*, *Port Prototype*, *Referrable*
Subclasses	PPortPrototype, PRPortPrototype

▽

△

Class	**AbstractProvidedPortPrototype** (abstract)			
Aggregated by	*AtpClassifier*.atpFeature, *SwComponentType*.port			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
providedCom Spec	PPortComSpec	*	aggr	Provided communication attributes per interface element (data element or operation).

Table A.13: AbstractProvidedPortPrototype

Class	**AbstractRequiredPortPrototype** (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components			
Note	This abstract class provides the ability to become a required PortPrototype.			
Base	*ARObject*, *AtpBlueprintable*, *AtpFeature*, *AtpPrototype*, *Identifiable*, *MultilanguageReferrable*, *Port Prototype*, *Referrable*			
Subclasses	PRPortPrototype, RPortPrototype			
Aggregated by	*AtpClassifier*.atpFeature, *SwComponentType*.port			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
requiredCom Spec	RPortComSpec	*	aggr	Required communication attributes, one for each interface element.

Table A.14: AbstractRequiredPortPrototype

Class	**AbstractRuleBasedValueSpecification** (abstract)			
Package	M2::AUTOSARTemplates::CommonStructure::Constants			
Note	This represents an abstract base class for all rule-based value specifications.			
Base	*ARObject*, *ValueSpecification*			
Subclasses	ApplicationRuleBasedValueSpecification, CompositeRuleBasedValueSpecification, NumericalRuleBased ValueSpecification			
Aggregated by	ApplicationAssocMapElementValueSpecification.key, ApplicationAssocMapElementValueSpecification. value, ArrayValueSpecification.element, CalibrationParameterValue.applInitValue, CalibrationParameter Value.implInitValue, ConstantSpecification.valueSpec, CryptoServiceKey.developmentValue, Diagnostic EnvDataCondition.compareValue, DiagnosticEnvDataElementCondition.compareValue, FieldSenderCom Spec.initValue, ISignal.initValue, ISignal.timeoutSubstitutionValue, NonqueuedReceiverComSpec.init Value, NonqueuedReceiverComSpec.timeoutSubstitutionValue, NonqueuedSenderComSpec.initValue, NvProvideComSpec.ramBlockInitValue, NvProvideComSpec.romBlockInitValue, NvRequireComSpec.init Value, ParameterDataPrototype.initValue, ParameterProvideComSpec.initValue, ParameterRequireCom Spec.initValue, PersistencyDataRequiredComSpec.initValue, PersistencyKeyValuePair.initValue, Port DefinedArgumentValue.value, PortPrototypeBlueprintInitValue.value, RecordValueSpecification.field, *StateManagementCompareCondition*.compareValue, SwDataDefProps.invalidValue, VariableData Prototype.initValue			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.15: AbstractRuleBasedValueSpecification

Class	**AbstractServiceInstance** (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances			
Note	Provided and Consumed Ethernet Service Instances that are available at the ApplicationEndpoint.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	ConsumedServiceInstance, ProvidedServiceInstance			
Aggregated by	ServiceInstanceCollectionSet.serviceInstance			

▽

△

Class	AbstractServiceInstance (abstract)			
Attribute	Type	Mult.	Kind	Note
capability Record	TagWithOptionalValue	*	aggr	A sequence of records to store arbitrary name/value pairs conveying additional information about the named service. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=capabilityRecord, capabilityRecord.variation Point.shortLabel vh.latestBindingTime=postBuild
majorVersion	PositiveInteger	0..1	attr	Major Version of the ServiceInterface. Value can be set to a number that represents the Major Version of the service.
method Activation RoutingGroup	PduActivationRouting Group	0..1	aggr	The ServiceDiscovery module is able to activate and deactivate the PDU routing for ClientServerOperations (SOME/IP methods). **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=methodActivationRoutingGroup.shortName, methodActivationRoutingGroup.variationPoint.shortLabel vh.latestBindingTime=postBuild
routingGroup	SoAdRoutingGroup	*	ref	The ServiceDiscovery module is able to activate and deactivate the PDU routing from and to TCP/IP-sockets. **Tags:**atp.Status=obsolete

Table A.16: AbstractServiceInstance

Class	AccessCount			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::AccessCount			
Note	This meta-class provides one count value for a AbstractAccessPoint.			
Base	ARObject			
Aggregated by	AccessCountSet.accessCount			
Attribute	Type	Mult.	Kind	Note
accessPoint	AbstractAccessPoint	0..1	ref	AbstractAccessPoint for which the count value is applicable.
value	PositiveInteger	0..1	attr	This attribute represents the number of determined accesses **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime

Table A.17: AccessCount

Class	AccessCountSet			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::AccessCount			
Note	This meta-class provides a set of count values evaluated according to the rules of a specific countProfile.			
Base	ARObject			
Aggregated by	ResourceConsumption.accessCountSet			
Attribute	Type	Mult.	Kind	Note

▽

△

Class	AccessCountSet			
accessCount	AccessCount	*	aggr	Count value for a AbstractAccessPoint. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=accessCount, accessCount.variation Point.shortLabel vh.latestBindingTime=preCompileTime
countProfile	NameToken	0..1	attr	This attribute defines the name of the count profile used to determine the AccessCount.value numbers.

Table A.18: AccessCountSet

Class	AdminData			
Package	M2::MSR::AsamHdo::AdminData			
Note	AdminData represents the ability to express administrative information and custom extensions for an element. This administration information is to be treated as meta-data such as revision id or state of the file. There are basically the following kinds of meta-data • The language and/or used languages. • Revision information covering e.g. revision number, state, release date, changes. Note that this information can be given in general as well as related to a particular company. • Document meta-data specific for a company Beside that a custom extension of model-data is possible by • Special data			
Base	ARObject			
Aggregated by	AUTOSAR.adminData, Describable.adminData, Identifiable.adminData			
Attribute	Type	Mult.	Kind	Note
docRevision (ordered)	DocRevision	*	aggr	This allows to denote information about the current revision of the object. Note that information about previous revisions can also be logged here. The entries shall be sorted descendant by date in order to reflect the history. Therefore the most recent entry representing the current version is denoted first. **Tags:** xml.roleElement=true xml.roleWrapperElement=true xml.sequenceOffset=50 xml.typeElement=false xml.typeWrapperElement=false
language	LEnum	0..1	attr	This attribute specifies the master language of the document or the document fragment. The master language is the one in which the document is maintained and from which the other languages are derived from. In particular in case of inconsistencies, the information in the master language is priority. **Tags:**xml.sequenceOffset=20

▽

△

Class	AdminData			
sdg	Sdg	*	aggr	This property allows to keep special data which is not represented by the standard model. It can be utilized to keep e.g. tool specific data. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=sdg xml.roleElement=true xml.roleWrapperElement=true xml.sequenceOffset=60 xml.typeElement=false xml.typeWrapperElement=false
usedLanguages	MultiLanguagePlainText	0..1	aggr	This property specifies the languages which are provided in the document. Therefore it should only be specified in the top level admin data. For each language provided in the document there is one entry in MultilanguagePlain Text. The content of each entry can be used for illustration of the language. The used language itself depends on the language attribute in the entry. **Tags:**xml.sequenceOffset=30

Table A.19: AdminData

Class	AgeConstraint			
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint::AgeConstraint			
Note	The AgeConstraint is used to impose a constraint on an Timing Description Event referenced by the scope. A minimum and a maximum age can be specified.			
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 age.
minimum	MultidimensionalTime	0..1	aggr	The minimum age.
scope	TimingDescriptionEvent	0..1	ref	The scope of an AgeConstraint is any TimingDescription Event that indicates any receipt of data.

Table A.20: AgeConstraint

Class	AggregationCondition			
Package	M2::AUTOSARTemplates::CommonStructure::StandardizationTemplate::DataExchangePoint::Data FormatTailoring			
Note	The AggregationCondition evaluates to true, if the referenced aggregation is accepted by all rules of this condition.			
Base	*ARObject*, *AbstractCondition*, *AbstractMultiplicityRestriction*, *AttributeCondition*			
Aggregated by	ClassContentConditional.condition, InvertCondition.condition			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
aggregation	AggregationTailoring	1	ref	The aggregation that has to be accepted by the restrictions of this AggregationCondition

Table A.21: AggregationCondition

Class	AggregationTailoring			
Package	M2::AUTOSARTemplates::CommonStructure::StandardizationTemplate::DataExchangePoint::Data FormatTailoring			
Note	Tailoring of aggregations in the AUTOSAR meta-model			
Base	*ARObject*, *AttributeTailoring*, *DataFormatElementReference*, *DataFormatElementScope*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *SpecElementReference*, *SpecElementScope*			
Aggregated by	ClassContentConditional.attributeTailoring			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
typeTailoring	ClassTailoring	*	aggr	Local class tailoring which is applied if the content is contained by this aggregation.

Table A.22: AggregationTailoring

Class	AliasNameAssignment			
Package	M2::AUTOSARTemplates::CommonStructure::FlatMap			
Note	This meta-class represents the ability to associate an alternative name to a flat representations or an Identifiable. The usage of this name is defined outside of AUTOSAR. For example this name can be used by MCD tools or as a name for component instances in the ECU extract. Note that flatInstance and identifiable are mutually exclusive.			
Base	*ARObject*			
Aggregated by	AliasNameSet.aliasName			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
flatInstance	FlatInstanceDescriptor	0..1	ref	Assignment of a unique name to a flat representation. **Tags:**xml.sequenceOffset=60
identifiable	Identifiable	0..1	ref	Assignment of a unique name to an Identifiable. **Tags:**xml.sequenceOffset=50
label	MultilanguageLong Name	0..1	aggr	This represents an "Alias LongName". **Tags:**xml.sequenceOffset=20
shortLabel	String	1	attr	This attribute represents the alias name. It is modeled as string because the alias name is used outside of AUTOSAR and therefore no naming conventions can be applied within AUTOSAR. **Stereotypes:** atpIdentityContributor **Tags:**xml.sequenceOffset=10

Table A.23: AliasNameAssignment

Class	AliasNameSet			
Package	M2::AUTOSARTemplates::CommonStructure::FlatMap			
Note	This meta-class represents a set of AliasNames. The AliasNameSet can for example be an input to the A2L-Generator. **Tags:**atp.recommendedPackage=AliasNameSets			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpBlueprintable*, *CollectableElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	AliasNameSet			
aliasName	AliasNameAssignment	*	aggr	AliasNames contained in the AliasNameSet. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=aliasName.shortLabel, aliasName.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table A.24: AliasNameSet

Class	AnalyzedExecutionTime			
Package	M2::AUTOSARTemplates::CommonStructure::ResourceConsumption::ExecutionTime			
Note	AnalyzedExecutionTime provides an analytic method for specifying the best and worst case execution time.			
Base	ARObject, ExecutionTime, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	ResourceConsumption.executionTime			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
bestCase ExecutionTime	MultidimensionalTime	0..1	aggr	The best case execution time (BCET) defines the minimum amount of time the related executable entity requires for its execution.
worstCase ExecutionTime	MultidimensionalTime	0..1	aggr	The worst case execution time (WCET) defines the maximum amount of time the related executable entity requires for its execution.

Table A.25: AnalyzedExecutionTime

Class	AnyInstanceRef			
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::AnyInstanceRef			
Note	Describes a reference to any instance in an AUTOSAR model. This is the most generic form of an instance ref. Refer to the superclass notes for more details.			
Base	ARObject, AtpInstanceRef			
Aggregated by	Collection.collectedInstance, Collection.sourceInstance, DocumentationContext.feature, EcucInstanceReferenceValue.value, FlatInstanceDescriptor.ecuExtractReference, FlatInstanceDescriptor.upstreamReference, RptContainer.byPassPoint, RptHook.rptArHook, ViewMap.firstElementInstance, ViewMap.secondElementInstance			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
base	AtpClassifier	1	ref	This is the base from which navigation path begins. **Stereotypes:** atpDerived
contextElement (ordered)	AtpFeature	*	ref	This is one step in the navigation path specified by the instance ref.
target	AtpFeature	1	ref	This is the target of the instance ref.

Table A.26: AnyInstanceRef

Primitive	AnyServiceInstanceId	
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::PrimitiveTypes	
Note	This is a positive integer or the literal ALL (the value ANY is technically supported but deprecated) which can be denoted in decimal, octal and hexadecimal. The value is between 0 and 65535. **Tags:** xml.xsd.customType=ANY-SERVICE-INSTANCE-ID xml.xsd.pattern=[1-9][0-9]*|0[xX][0-9a-fA-F]+|0[0-7]*|0[bB][0-1]+|ANY|ALL xml.xsd.type=string	

Table A.27: AnyServiceInstanceId

Class	ApplicationArrayDataType			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::Datatypes			
Note	An application data type which is an array, each element is of the same application data type. **Tags:**atp.recommendedPackage=ApplicationDataTypes			
Base	*ARElement*, ARObject, *ApplicationCompositeDataType*, *ApplicationDataType*, *AtpBlueprint*, *Atp Blueprintable*, *AtpClassifier*, *AtpType*, *AutosarDataType*, CollectableElement, *Identifiable*, *Multilanguage Referrable*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
dynamicArray SizeProfile	String	0..1	attr	Specifies the profile which the array will follow if it is a variable size array.
element	ApplicationArray Element	0..1	aggr	This association implements the concept of an array element. That is, in some cases it is necessary to be able to identify single array elements, e.g. as input values for an interpolation routine.

Table A.28: ApplicationArrayDataType

Class	ApplicationArrayElement			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::DataPrototypes			
Note	Describes the properties of the elements of an application array data type.			
Base	ARObject, *ApplicationCompositeElementDataPrototype*, AtpFeature, *AtpPrototype*, *DataPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ApplicationArrayDataType.element, *AtpClassifier*.atpFeature			
Attribute	Type	Mult.	Kind	Note
arraySize Handling	ArraySizeHandling Enum	0..1	attr	The way how the size of the array is handled.
arraySize Semantics	ArraySizeSemantics Enum	0..1	attr	This attribute controls how the information about the array size shall be interpreted.
indexDataType	ApplicationPrimitive DataType	0..1	ref	This reference can be taken to assign a CompuMethod of category TEXTTABLE to the array. The texttable entries associate a textual value to an index number such that the element with that index number is represented by a symbolic name.
maxNumberOf Elements	PositiveInteger	0..1	attr	The maximum number of elements that the array can contain. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime

Table A.29: ApplicationArrayElement

Class	ApplicationCompositeDataType (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::Datatypes			
Note	Abstract base class for all application data types composed of other data types.			
Base	*ARElement*, ARObject, *ApplicationDataType*, *AtpBlueprint*, AtpBlueprintable, *AtpClassifier*, *AtpType*, *AutosarDataType*, CollectableElement, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	ApplicationArrayDataType, ApplicationRecordDataType			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.30: ApplicationCompositeDataType

Class	ApplicationCompositeDataTypeSubElementRef			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	This meta-class represents the specialization of SubElementMapping with respect to Application CompositeDataTypes.			
Base	*ARObject*, *SubElementRef*			
Aggregated by	SubElementMapping.firstElement, SubElementMapping.secondElement			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
application Composite Element	ApplicationComposite ElementDataPrototype	0..1	iref	This represents the referenced ApplicationComposite DataPrototype. **InstanceRef implemented by:** ApplicationComposite ElementInPortInterfaceInstanceRef

Table A.31: ApplicationCompositeDataTypeSubElementRef

Class	ApplicationCompositeElementDataPrototype (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::DataPrototypes			
Note	This class represents a data prototype which is aggregated within a composite application data type (record or array). It is introduced to provide a better distinction between target and context in instance Refs.			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *DataPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	ApplicationArrayElement, ApplicationRecordElement			
Aggregated by	*AtpClassifier*.atpFeature			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
type	ApplicationDataType	0..1	tref	This represents the corresponding data type. **Stereotypes:** isOfType

Table A.32: ApplicationCompositeElementDataPrototype

Class	ApplicationCompositeElementInPortInterfaceInstanceRef			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface::InstanceRefs			
Note				
Base	*ARObject*, *AtpInstanceRef*			
Aggregated by	ApplicationCompositeDataTypeSubElementRef.applicationCompositeElement, CompositeNetwork Representation.leafElement			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
base	DataInterface	0..1	ref	This represents the SenderReceiverInterface that acts as the base in this InstanceRef definition **Stereotypes:** atpDerived **Tags:** xml.sequenceOffset=10
contextData Prototype (ordered)	ApplicationComposite ElementDataPrototype	*	ref	This represents a context ApplicationCompositeData Prototype **Tags:** xml.sequenceOffset=20
rootData Prototype	AutosarDataPrototype	0..1	ref	This refers to the dataPrototype which is typed by the ApplicationDatatype in which which the target can be found. **Tags:** xml.sequenceOffset=15
targetData Prototype	ApplicationComposite ElementDataPrototype	0..1	ref	This represents the referenced ApplicationComposite DataPrototype. **Tags:** xml.sequenceOffset=30

Table A.33: ApplicationCompositeElementInPortInterfaceInstanceRef

Class	ApplicationDataType (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::Datatypes			
Note	ApplicationDataType defines a data type from the application point of view. Especially it should be used whenever something "physical" is at stake. An ApplicationDataType represents a set of values as seen in the application model, such as measurement units. It does not consider implementation details such as bit-size, endianess, etc. It should be possible to model the application level aspects of a VFB system by using ApplicationData Types only.			
Base	ARElement, ARObject, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, AutosarDataType, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Subclasses	ApplicationCompositeDataType, ApplicationPrimitiveDataType			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.34: ApplicationDataType

Class	ApplicationEndpoint			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	An application endpoint is the endpoint on an Ecu in terms of application addressing (e.g. socket). The application endpoint represents e.g. the listen socket in client-server-based communication.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	SocketAddress.applicationEndpoint			
Attribute	Type	Mult.	Kind	Note
consumed ServiceInstance	ConsumedService Instance	*	aggr	Consumed service instances. **Tags:**atp.Status=obsolete
maxNumberOf Connections	PositiveInteger	0..1	attr	This attribute defines the maximal number of clients the Server is able to deal with in case of Service Discovery.
network Endpoint	NetworkEndpoint	1	ref	Reference to the network address.
priority	PositiveInteger	0..1	attr	Defines the frame priority where values from 0 (best effort) to 7 (highest) are allowed.
providedService Instance	ProvidedService Instance	*	aggr	Provided service instances. **Tags:**atp.Status=obsolete
tlsCrypto Mapping	TlsCryptoService Mapping	0..1	ref	This reference identifies the applicable TlsCryptoService Mapping that adds the ability for TLS-based encryption on the enclosing ApplicationEndpoint.
tpConfiguration	TransportProtocol Configuration	0..1	aggr	Configuration of the used transport protocol.

Table A.35: ApplicationEndpoint

Class	ApplicationError			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	This is a user-defined error that is associated with an element of an AUTOSAR interface. It is specific for the particular functionality or service provided by the AUTOSAR software component.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	ClientServerInterface.possibleError			
Attribute	Type	Mult.	Kind	Note

▽

△

Class	ApplicationError			
errorCode	Integer	0..1	attr	The RTE generator is forced to assign this value to the corresponding error symbol. Note that for error codes certain ranges are predefined (see RTE specification).

Table A.36: ApplicationError

Class	ApplicationPartition			
Package	M2::AUTOSARTemplates::SystemTemplate::SWmapping			
Note	ApplicationPartition to which SwComponentPrototypes are mapped at a point in time when the corresponding EcuInstance is not yet known or defined. In a later methodology step the Application Partition can be assigned to an EcuPartition. **Tags:**atp.recommendedPackage=ApplicationPartitions			
Base	*ARElement*, ARObject, CollectableElement, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.37: ApplicationPartition

Class	ApplicationPartitionToEcuPartitionMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::SWmapping			
Note	Maps ApplicationPartitions to EcuPartitions. With this mapping an OEM has the option to predefine an allocation of Software Components to EcuPartitions in the System Design phase. The final and complete assignment is described in the OS Configuration.			
Base	ARObject, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SystemMapping.applicationPartitionToEcuPartitionMapping			
Attribute	Type	Mult.	Kind	Note
application Partition	ApplicationPartition	*	ref	Reference to ApplicationPartitions that are mapped to an EcuPartition.
ecuPartition	EcuPartition	0..1	ref	Reference to EcuPartition to which the Application Partitions are assigned.

Table A.38: ApplicationPartitionToEcuPartitionMapping

Class	ApplicationPrimitiveDataType			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::Datatypes			
Note	A primitive data type defines a set of allowed values. **Tags:**atp.recommendedPackage=ApplicationDataTypes			
Base	*ARElement*, ARObject, *ApplicationDataType*, *AtpBlueprint*, AtpBlueprintable, *AtpClassifier*, *AtpType*, *AutosarDataType*, CollectableElement, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.39: ApplicationPrimitiveDataType

Class	ApplicationRecordDataType			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::Datatypes			
Note	An application data type which can be decomposed into prototypes of other application data types. **Tags:**atp.recommendedPackage=ApplicationDataTypes			
Base	*ARElement*, ARObject, *ApplicationCompositeDataType*, *ApplicationDataType*, *AtpBlueprint*, *Atp Blueprintable*, *AtpClassifier*, *AtpType*, *AutosarDataType*, CollectableElement, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
element (ordered)	ApplicationRecord Element	*	aggr	Specifies an element of a record. The aggregation of ApplicationRecordElement is subject to variability with the purpose to support the conditional existence of elements inside a ApplicationrecordData Type. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=element.shortName, element.variation Point.shortLabel vh.latestBindingTime=preCompileTime

Table A.40: ApplicationRecordDataType

Class	ApplicationRecordElement			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::DataPrototypes			
Note	Describes the properties of one particular element of an application record data type.			
Base	ARObject, *ApplicationCompositeElementDataPrototype*, *AtpFeature*, *AtpPrototype*, *DataPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ApplicationRecordDataType.element, *AtpClassifier*.atpFeature			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
isOptional	Boolean	0..1	attr	This attribute represents the ability to declare the enclosing ApplicationRecordElement as optional. This means the that, at runtime, the ApplicationRecord Element may or may not have a valid value and shall therefore be ignored. The underlying runtime software provides means to set the ApplicationRecordElement as not valid at the sending end of a communication and determine its validity at the receiving end.

Table A.41: ApplicationRecordElement

Class	ApplicationRuleBasedValueSpecification
Package	M2::AUTOSARTemplates::CommonStructure::Constants
Note	This meta-class represents rule based values for DataPrototypes typed by ApplicationDataTypes (ApplicationArrayDataType or a compound ApplicationPrimitiveDataType which also boils down to an array-nature).
Base	ARObject, *AbstractRuleBasedValueSpecification*, *CompositeRuleBasedValueArgument*, *Value Specification*

▽

△

Class	ApplicationRuleBasedValueSpecification			
Aggregated by	ApplicationAssocMapElementValueSpecification.key, ApplicationAssocMapElementValueSpecification. value, ArrayValueSpecification.element, CalibrationParameterValue.applInitValue, CalibrationParameter Value.implInitValue, CompositeRuleBasedValueSpecification.compoundPrimitiveArgument, Constant Specification.valueSpec, CryptoServiceKey.developmentValue, DiagnosticEnvDataCondition.compare Value, DiagnosticEnvDataElementCondition.compareValue, FieldSenderComSpec.initValue, ISignal.init Value, ISignal.timeoutSubstitutionValue, NonqueuedReceiverComSpec.initValue, NonqueuedReceiver ComSpec.timeoutSubstitutionValue, NonqueuedSenderComSpec.initValue, NvProvideComSpec.ram BlockInitValue, NvProvideComSpec.romBlockInitValue, NvRequireComSpec.initValue, ParameterData Prototype.initValue, ParameterProvideComSpec.initValue, ParameterRequireComSpec.initValue, PersistencyDataRequiredComSpec.initValue, PersistencyKeyValuePair.initValue, PortDefinedArgument Value.value, PortPrototypeBlueprintInitValue.value, RecordValueSpecification.field, *StateManagement CompareCondition*.compareValue, SwDataDefProps.invalidValue, VariableDataPrototype.initValue			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
category	Identifier	0..1	attr	This represents the category of the RuleBasedValue Specification **Tags:**xml.sequenceOffset=-20
swAxisCont (ordered)	RuleBasedAxisCont	*	aggr	This represents the axis values of a Compound Primitive Data Type (curve or map). The first swAxisCont describes the x-axis, the second sw AxisCont describes the y-axis, the third swAxisCont describes the z-axis. In addition to this, the axis can be denoted in swAxisIndex.
swValueCont	RuleBasedValueCont	0..1	aggr	This represents the values of an array or Compound Primitive Data Type.

Table A.42: ApplicationRuleBasedValueSpecification

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*, *Atp Type*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *Sw ComponentType*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.43: ApplicationSwComponentType

Class	ApplicationValueSpecification
Package	M2::AUTOSARTemplates::CommonStructure::Constants
Note	This meta-class represents values for DataPrototypes typed by ApplicationDataTypes (this includes in particular compound primitives). For further details refer to ASAM CDF 2.0. This meta-class corresponds to some extent with SW-INSTANCE in ASAM CDF 2.0.
Base	*ARObject*, *CompositeRuleBasedValueArgument*, *ValueSpecification*

▽

△

Class	ApplicationValueSpecification			
Aggregated by	ApplicationAssocMapElementValueSpecification.key, ApplicationAssocMapElementValueSpecification.value, ArrayValueSpecification.element, CalibrationParameterValue.applInitValue, CalibrationParameterValue.implInitValue, CompositeRuleBasedValueSpecification.compoundPrimitiveArgument, ConstantSpecification.valueSpec, CryptoServiceKey.developmentValue, DiagnosticEnvDataCondition.compareValue, DiagnosticEnvDataElementCondition.compareValue, FieldSenderComSpec.initValue, ISignal.initValue, ISignal.timeoutSubstitutionValue, NonqueuedReceiverComSpec.initValue, NonqueuedReceiverComSpec.timeoutSubstitutionValue, NonqueuedSenderComSpec.initValue, NvProvideComSpec.ramBlockInitValue, NvProvideComSpec.romBlockInitValue, NvRequireComSpec.initValue, ParameterDataPrototype.initValue, ParameterProvideComSpec.initValue, ParameterRequireComSpec.initValue, PersistencyDataRequiredComSpec.initValue, PersistencyKeyValuePair.initValue, PortDefinedArgumentValue.value, PortPrototypeBlueprintInitValue.value, RecordValueSpecification.field, *StateManagementCompareCondition*.compareValue, SwDataDefProps.invalidValue, VariableDataPrototype.initValue			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
category	Identifier	0..1	attr	Specifies to which category of ApplicationDataType this ApplicationValueSpecification can be applied (e.g. as an initial value), thus imposing constraints on the structure and semantics of the contained values, see [constr_1006] and [constr_2051].
swAxisCont (ordered)	SwAxisCont	*	aggr	This represents the axis values of a Compound Primitive Data Type (curve or map). The first swAxisCont describes the x-axis, the second swAxisCont describes the y-axis, the third swAxisCont describes the z-axis. In addition to this, the axis can be denoted in swAxisIndex.
swValueCont	SwValueCont	0..1	aggr	This represents the values of a Compound Primitive Data Type.

Table A.44: ApplicationValueSpecification

Class	ArParameterInImplementationDataInstanceRef			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::DataElements			
Note	This class represents the ability to navigate into an element inside of an ParameterDataPrototype typed by an ImplementationDatatype. Note that it shall not be used if the target is the ParameterDataPrototype itself (e.g. if the target is a primitive data type). Note that this class follows the pattern of an InstanceRef but is not implemented based on the abstract classes because the ImplementationDataType isn't either, especially because ImplementationDataTypeElement (intentionally) isn't derived from AtpPrototype.			
Base	*ARObject*			
Aggregated by	ImplementationDataTypeSubElementRef.parameterImplementationDataTypeElement			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
contextDataPrototype (ordered)	AbstractImplementationDataTypeElement	*	ref	This is a context in case there are subelements with explicit types. The reference has to be ordered to properly reflect the nested structure.
portPrototype	PortPrototype	0..1	ref	This reference points to the PortPrototype providing/receiving the root of the parameter.
rootParameterDataPrototype	ParameterDataPrototype	0..1	ref	This refers to the ParameterDataPrototype typed by the implementationDataType in which the target can be found.
targetDataPrototype	AbstractImplementationDataTypeElement	0..1	ref	This reference points to the target ImplementationDataTypeElement.

Table A.45: ArParameterInImplementationDataInstanceRef

Class	ArVariableInImplementationDataInstanceRef			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::DataElements			
Note	This class represents the ability to navigate into a data element inside of an VariableDataPrototype which is typed by an ImplementationDatatype. Note that it shall not be used if the target is the VariableDataPrototype itself (e.g. if its a primitive). Note that this class follows the pattern of an InstanceRef but is not implemented based on the abstract classes because the ImplementationDataType isn't either, especially because ImplementationDataType Element isn't derived from AtpPrototype.			
Base	ARObject			
Aggregated by	AutosarVariableRef.autosarVariableInImplDatatype, ImplementationDataTypeSubElementRef. implementationDataTypeElement			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
contextData Prototype (ordered)	AbstractImplementation DataTypeElement	*	ref	This is a context in case there are subelements with explicit types. The reference has to be ordered to properly reflect the nested structure. **Tags:**xml.sequenceOffset=30
portPrototype	PortPrototype	0..1	ref	This is the port providing/receiving the root of the variable **Tags:**xml.sequenceOffset=10
rootVariable DataPrototype	VariableDataPrototype	0..1	ref	This refers to the VariableDataPrototype typed by the ImplementationDatatype in which the target can be found. **Tags:**xml.sequenceOffset=20
targetData Prototype	AbstractImplementation DataTypeElement	0..1	ref	This reference points to the target ImplementationData TypeElement. **Tags:**xml.sequenceOffset=40

Table A.46: ArVariableInImplementationDataInstanceRef

Class	ArbitraryEventTriggering			
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint::EventTriggeringConstraint			
Note	The ArbitraryEventTriggering describes that an event occurs occasionally, singly, irregularly or randomly. 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.			
Base	ARObject, EventTriggeringConstraint, Identifiable, MultilanguageReferrable, Referrable, Timing Constraint, Traceable			
Aggregated by	TimingExtension.timingGuarantee, TimingExtension.timingRequirement			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
confidence Interval	ConfidenceInterval	*	aggr	List of confidence intervals. **Tags:**xml.sequenceOffset=30
maximum Distance	MultidimensionalTime	*	aggr	The nth array element describes the maximum distance that can be observed for a sample of n+1 event occurrences. This is an array with an identical number of elements as for the minimumDistance. **Tags:** xml.name=TIME-VALUE xml.roleElement=true xml.sequenceOffset=20 xml.typeElement=false

$\bigtriangledown$

△

Class	ArbitraryEventTriggering			
minimum Distance	MultidimensionalTime	*	aggr	The nth array element describes the minimum distance that can be observed for a sample of n+1 event occurrences. This is an array with an identical number of elements as for the maximumDistance. **Tags:** xml.name=TIME-VALUE xml.roleElement=true xml.sequenceOffset=10 xml.typeElement=false

Table A.47: ArbitraryEventTriggering

Class	ArgumentDataPrototype			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	An argument of an operation, much like a data element, but also carries direction information and is owned by a particular ClientServerOperation.			
Base	ARObject, AtpFeature, AtpPrototype, AutosarDataPrototype, DataPrototype, Identifiable, Multilanguage Referrable, Referrable			
Aggregated by	AtpClassifier.atpFeature, ClientServerOperation.argument			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
direction	ArgumentDirection Enum	0..1	attr	This attribute specifies the direction of the argument prototype.
serverArgument ImplPolicy	ServerArgumentImpl PolicyEnum	0..1	attr	This defines how the argument type of the servers RunnableEntity is implemented. If the attribute is not defined this has the same semantics as if the attribute is set to the value useArgumentType for primitive arguments and structures.

Table A.48: ArgumentDataPrototype

Enumeration	ArgumentDirectionEnum
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::PrimitiveTypes
Note	Use cases: • Arguments in ClientServerOperation can have different directions that need to be formally indicated because they have an impact on how the function signature looks like eventually. • Arguments in BswModuleEntry already determine a function signature, but the direction is used to specify the semantics, especially of pointer arguments.
Aggregated by	ArgumentDataPrototype.direction, SwServiceArg.direction
Literal	*Description*
in	The argument value is passed to the callee. **Tags:**atp.EnumerationLiteralIndex=0
inout	The argument value is passed to the callee but also passed back from the callee to the caller. **Tags:**atp.EnumerationLiteralIndex=1
out	The argument value is passed from the callee to the caller. **Tags:**atp.EnumerationLiteralIndex=2

Table A.49: ArgumentDirectionEnum

Enumeration	ArraySizeHandlingEnum
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::Datatypes
Note	This enumeration defines different ways to handle the sizes of variable size arrays.
Aggregated by	ApplicationArrayElement.arraySizeHandling, ImplementationDataTypeElement.arraySizeHandling
Literal	Description
allIndicesDifferent ArraySize	All elements of the variable size array may have different sizes. **Tags:**atp.EnumerationLiteralIndex=0
allIndicesSame ArraySize	All elements of the variable size array have the same size. **Tags:**atp.EnumerationLiteralIndex=1
inheritedFromArray ElementTypeSize	The size of all dimensions of the variable size array is determined by the size of the contained array element. **Tags:**atp.EnumerationLiteralIndex=2

Table A.50: ArraySizeHandlingEnum

Enumeration	ArraySizeSemanticsEnum
Package	M2::AUTOSARTemplates::CommonStructure::ImplementationDataTypes
Note	This type controls how the information about the number of elements in an ApplicationArrayDataType is to be interpreted.
Aggregated by	ApplicationArrayElement.arraySizeSemantics, DiagnosticDataElement.arraySizeSemantics, ImplementationDataTypeElement.arraySizeSemantics, SwTextProps.arraySizeSemantics
Literal	Description
fixedSize	This means that the ApplicationArrayDataType will always have a fixed number of elements. **Tags:**atp.EnumerationLiteralIndex=0
variableSize	This implies that the actual number of elements in the ApplicationArrayDataType might vary at run-time. The value of arraySize represents the maximum number of elements in the array. **Tags:**atp.EnumerationLiteralIndex=1

Table A.51: ArraySizeSemanticsEnum

Class	ArrayValueSpecification
Package	M2::AUTOSARTemplates::CommonStructure::Constants
Note	Specifies the values for an array.
Base	ARObject, CompositeValueSpecification, ValueSpecification
Aggregated by	ApplicationAssocMapElementValueSpecification.key, ApplicationAssocMapElementValueSpecification. value, ArrayValueSpecification.element, CalibrationParameterValue.applInitValue, CalibrationParameter Value.implInitValue, CompositeRuleBasedValueSpecification.argument, ConstantSpecification.value Spec, CryptoServiceKey.developmentValue, DiagnosticEnvDataCondition.compareValue, DiagnosticEnv DataElementCondition.compareValue, FieldSenderComSpec.initValue, ISignal.initValue, ISignal.timeout SubstitutionValue, NonqueuedReceiverComSpec.initValue, NonqueuedReceiverComSpec.timeout SubstitutionValue, NonqueuedSenderComSpec.initValue, NvProvideComSpec.ramBlockInitValue, Nv ProvideComSpec.romBlockInitValue, NvRequireComSpec.initValue, ParameterDataPrototype.initValue, ParameterProvideComSpec.initValue, ParameterRequireComSpec.initValue, PersistencyDataRequired ComSpec.initValue, PersistencyKeyValuePair.initValue, PortDefinedArgumentValue.value, PortPrototype BlueprintInitValue.value, RecordValueSpecification.field, StateManagementCompareCondition.compare Value, SwDataDefProps.invalidValue, VariableDataPrototype.initValue
Attribute	Type	Mult.	Kind	Note

▽

△

Class	ArrayValueSpecification			
element (ordered)	ValueSpecification	*	aggr	The value for a single array element. All Value Specifications aggregated by ArrayValueSpecification shall have the same structure. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=element, element.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
intendedPartial Initialization Count	PositiveInteger	0..1	attr	This attribute shall only have a meaning for dynamic arrays and shall be taken as a sanity check: the number filled in the attribute shall be identical to the number of ArrayValueSpecification.element. If the attribute does not exist it means that no partial initialization is intended.

Table A.52: ArrayValueSpecification

Class	AssemblySwConnector			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Composition			
Note	AssemblySwConnectors are exclusively used to connect SwComponentPrototypes in the context of a CompositionSwComponentType.			
Base	ARObject, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable, MultilanguageReferrable, Referrable, SwConnector			
Aggregated by	AtpClassifier.atpFeature, CompositionSwComponentType.connector			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
provider	AbstractProvidedPort Prototype	0..1	iref	Instance of providing port. **InstanceRef implemented by:**PPortInComposition InstanceRef
requester	AbstractRequiredPort Prototype	0..1	iref	Instance of requiring port. **InstanceRef implemented by:**RPortInComposition InstanceRef

Table A.53: AssemblySwConnector

Class	AssignFrameIdRange			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	AssignFrameIdRange generates an assign frame PID range request.			
Base	ARObject, LinConfigurationEntry, ScheduleTableEntry			
Aggregated by	LinScheduleTable.tableEntry			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
framePid	FramePid	0..4	aggr	Optional assignment of frame_PID values that are included in the request. The frame_PIDs are ordered.
startIndex	Integer	1	attr	The startIndex sets the index to the first frame to assign a PID.

Table A.54: AssignFrameIdRange

Class	AsynchronousServerCallPoint			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::ServerCall			
Note	An AsynchronousServerCallPoint is used for asynchronous invocation of a ClientServerOperation. IMPORTANT: a ServerCallPoint cannot be used concurrently. Once the client RunnableEntity has made the invocation, the ServerCallPoint cannot be used until the call returns (or an error occurs!) at which point the ServerCallPoint becomes available again.			
Base	ARObject, *AbstractAccessPoint*, *AtpClassifier*, AtpFeature, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *ServerCallPoint*			
Aggregated by	*AtpClassifier*.atpFeature, RunnableEntity.serverCallPoint			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.55: AsynchronousServerCallPoint

Class	AsynchronousServerCallResultPoint			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::ServerCall			
Note	If a RunnableEntity owns a AsynchronousServerCallResultPoint it is entitled to get the result of the referenced AsynchronousServerCallPoint. If it is associated with AsynchronousServerCallReturnsEvent, this RTEEvent notifies the completion of the required ClientServerOperation or a timeout. The occurrence of this event can either unblock a WaitPoint or can lead to the invocation of a RunnableEntity.			
Base	ARObject, *AbstractAccessPoint*, *AtpClassifier*, AtpFeature, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, RunnableEntity.asynchronousServerCallResultPoint			
Attribute	Type	Mult.	Kind	Note
asynchronous ServerCallPoint	AsynchronousServer CallPoint	0..1	ref	The referenced Asynchronous Server Call Point defines the asynchronous server call from which the results are returned.

Table A.56: AsynchronousServerCallResultPoint

Class	AsynchronousServerCallReturnsEvent			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents			
Note	This event is raised when an asynchronous server call is finished.			
Base	ARObject, *AbstractEvent*, *AtpClassifier*, AtpFeature, *AtpStructureElement*, *Identifiable*, *Multilanguage Referrable*, *RTEEvent*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, SwcInternalBehavior.event			
Attribute	Type	Mult.	Kind	Note
eventSource	AsynchronousServer CallResultPoint	0..1	ref	The referenced AsynchronousServerCallResultPoint raises this AsynchronousServerCallReturnsEvent when the asynchronous server call returns.

Table A.57: AsynchronousServerCallReturnsEvent

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*

▽

△

Class	**AtomicSwComponentType** (abstract)			
Subclasses	ApplicationSwComponentType, ComplexDeviceDriverSwComponentType, EcuAbstractionSwComponentType, NvBlockSwComponentType, SensorActuatorSwComponentType, ServiceProxySwComponentType, ServiceSwComponentType			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
internalBehavior	SwcInternalBehavior	0..1	aggr	The SwcInternalBehaviors owned by an AtomicSw ComponentType can be located in a different physical file. Therefore the aggregation is <<atpSplitable>>. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=internalBehavior.shortName, internal Behavior.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
symbolProps	SymbolProps	0..1	aggr	This represents the SymbolProps for the AtomicSw ComponentType. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=symbolProps.shortName

Table A.58: AtomicSwComponentType

Class	**AtpBlueprint** (abstract)			
Package	M2::AUTOSARTemplates::CommonStructure::StandardizationTemplate::AbstractBlueprintStructure			
Note	This meta-class represents the ability to act as a Blueprint. As this class is an abstract one, particular blueprint meta-classes inherit from this one.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Subclasses	ARPackage, AbstractImplementationDataType, AclObjectSet, AclOperation, AclPermission, AclRole, AliasNameSet, ApplicationDataType, BswEntryRelationshipSet, BswModuleDescription, BswModuleEntry, BuildActionEntity, BuildActionEnvironment, BuildActionManifest, ClientServerInterfaceToBswModuleEntryBlueprintMapping, CompuMethod, ConsistencyNeeds, DataConstr, DataTypeMappingSet, EcucDefinitionCollection, EcucDestinationUriDefSet, EcucModuleDef, FlatMap, KeywordSet, LifeCycleState, LifeCycleStateDefinitionGroup, ModeDeclarationGroup, PortInterface, PortInterfaceMapping, PortInterfaceMappingSet, PortPrototypeBlueprint, SwAddrMethod, SwBaseType, SwComponentType, VfbTiming			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
blueprintPolicy	BlueprintPolicy	*	aggr	This role indicates whether the blueprintable element will be modifiable or not modifiable.

Table A.59: AtpBlueprint

Class	**AtpBlueprintMapping** (abstract)			
Package	M2::AUTOSARTemplates::CommonStructure::StandardizationTemplate::AbstractBlueprintStructure			
Note	This meta-class represents the ability to express a particular mapping between a blueprint and an element derived from this blueprint. Particular mappings are defined by specializations of this meta-class.			
Base	ARObject			
Subclasses	BlueprintMapping			
Aggregated by	BlueprintMappingSet.blueprintMap			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	AtpBlueprintMapping (abstract)			
atpBlueprint	AtpBlueprint	1	ref	This represents the blueprint. **Stereotypes:** atpAbstract; atpUriDef **Tags:**xml.sequenceOffset=50
atpBlueprinted Element	AtpBlueprintable	1	ref	This represents the bluprinted elements which shall be mapped to the blueprint. **Stereotypes:** atpAbstract **Tags:**xml.sequenceOffset=60

Table A.60: AtpBlueprintMapping

Class	AtpClassifier (abstract)			
Package	M2::AUTOSARTemplates::GenericStructure::AbstractStructure			
Note	A classifier classifies M0 instances according to their features. Or: a classifier is something that has instances - an M1 classifier has M0 instances.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Subclasses	AtpStructureElement, AtpType			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
atpFeature	AtpFeature	*	aggr	This is a feature of the classifier. **Stereotypes:** atpDerived

Table A.61: AtpClassifier

Class	AtpInstanceRef (abstract)			
Package	M2::AUTOSARTemplates::GenericStructure::AbstractStructure			
Note	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. 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).			
Base	ARObject			
Subclasses	AnyInstanceRef, ApplicationCompositeElementInPortInterfaceInstanceRef, ComponentInComposition InstanceRef, ComponentInSystemInstanceRef, DataPrototypeInPortInterfaceInstanceRef, Data PrototypeInSystemInstanceRef, InnerDataPrototypeGroupInCompositionInstanceRef, InnerPortGroupIn CompositionInstanceRef, InnerRunnableEntityGroupInCompositionInstanceRef, InstanceEventIn CompositionInstanceRef, ModeDeclarationGroupPrototypeInSystemInstanceRef, ModeGroupInAtomic SwcInstanceRef, ModeInBswModuleDescriptionInstanceRef, ModeInSwcInstanceRef, Operation ArgumentInComponentInstanceRef, OperationInAtomicSwcInstanceRef, OperationInSystemInstance Ref, PModeInSystemInstanceRef, ParameterDataPrototypeInSystemInstanceRef, ParameterInAtomicS WCTypeInstanceRef, PortGroupInSystemInstanceRef, PortInCompositionTypeInstanceRef, RModeIn AtomicSwcInstanceRef, RteEventInCompositionInstanceRef, RteEventInEcuInstanceRef, RteEventIn SystemInstanceRef, RunnableEntityInCompositionInstanceRef, SwcServiceDependencyInSystem InstanceRef, TriggerInAtomicSwcInstanceRef, TriggerInSystemInstanceRef, VariableAccessInEcu InstanceRef, VariableDataPrototypeInCompositionInstanceRef, VariableDataPrototypeInSystemInstance Ref, VariableInAtomicSWCTypeInstanceRef, VariableInAtomicSwcInstanceRef, VariableInComponent InstanceRef			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
atpBase	AtpClassifier	1	ref	This is the base from which the navigaion path starts. **Stereotypes:** atpAbstract; atpDerived
atpContext Element (ordered)	AtpPrototype	*	ref	This is one particular step in the navigation path. **Stereotypes:** atpAbstract

▽

△

Class	AtpInstanceRef (abstract)			
atpTarget	AtpFeature	1	ref	This is the target of the instance ref. In other words it is the terminal of the navigation path. **Stereotypes:** atpAbstract

Table A.62: AtpInstanceRef

Class	AtpPrototype (abstract)			
Package	M2::AUTOSARTemplates::GenericStructure::AbstractStructure			
Note	A prototype is a typed feature. A prototype in a classifier indicates that instances of that classifier will have a feature, and the structure of that feature is given by the its type. An instance of that type will play the role indicated by the feature in the owning classifier. A feature is not an instance but an indication of an instance-to-be.			
Base	ARObject, AtpFeature, Identifiable, MultilanguageReferrable, Referrable			
Subclasses	DataPrototype, ModeDeclarationGroupPrototype, PortPrototype, RootSwCompositionPrototype, SwComponentPrototype			
Aggregated by	AtpClassifier.atpFeature			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
atpType	AtpType	1	ref	This is the type of the feature. **Stereotypes:** atpAbstract

Table A.63: AtpPrototype

Class	AtpStructureElement (abstract)			
Package	M2::AUTOSARTemplates::GenericStructure::AbstractStructure			
Note	A structure element is both a classifier and a feature. As a feature, its structure is given by the feature it owns as a classifier.			
Base	ARObject, AtpClassifier, AtpFeature, Identifiable, MultilanguageReferrable, Referrable			
Subclasses	AbstractAccessPoint, AbstractImplementationDataTypeElement, AsynchronousServerCallResultPoint, BswModuleDescription, BulkNvDataDescriptor, ClientServerOperation, DataPrototypeGroup, Ident Caption, InternalBehavior, InternalTriggeringPoint, ModeDeclaration, ModeDeclarationMapping, Mode SwitchPoint, ModeTransition, NvBlockDescriptor, ParameterAccess, PerInstanceMemory, PortGroup, PortPrototypeBlueprint, RTEEvent, RunnableEntity, RunnableEntityGroup, ServerCallPoint, Sw Connector, SwcBswMapping, SwcServiceDependency, System, Trigger, VariableAccess			
Aggregated by	AtpClassifier.atpFeature			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.64: AtpStructureElement

Class	AtpType (abstract)			
Package	M2::AUTOSARTemplates::GenericStructure::AbstractStructure			
Note	A type is a classifier that may serve to type prototypes. It is a reusable classifier.			
Base	ARObject, AtpClassifier, Identifiable, MultilanguageReferrable, Referrable			
Subclasses	AutosarDataType, ModeDeclarationGroup, ModeDeclarationMappingSet, PortInterface, SwComponentType			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.65: AtpType

Class	**AttributeTailoring** (abstract)			
Package	M2::AUTOSARTemplates::CommonStructure::StandardizationTemplate::DataExchangePoint::Data FormatTailoring			
Note	Tailoring of Attributes			
Base	*ARObject*, *DataFormatElementReference*, *DataFormatElementScope*, *Identifiable*, *Multilanguage Referrable*, *Referrable*, *SpecElementReference*, *SpecElementScope*			
Subclasses	AggregationTailoring, PrimitiveAttributeTailoring, ReferenceTailoring			
Aggregated by	ClassContentConditional.attributeTailoring			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
multiplicity Restriction	MultiplicityRestriction WithSeverity	0..1	aggr	Multiplicity restriction of the attribute **Tags:**xml.sequenceOffset=10
variation Restriction	VariationRestrictionWith Severity	0..1	aggr	Restrictions on the usage of variant handling. **Tags:**xml.sequenceOffset=20

Table A.66: AttributeTailoring

Class	<<atpMixedString>> **AttributeValueVariationPoint** (abstract)			
Package	M2::AUTOSARTemplates::GenericStructure::VariantHandling::AttributeValueVariationPoints			
Note	This class represents the ability to derive the value of the Attribute from a system constant (by Sw SystemconstDependentFormula). It also provides a bindingTime.			
Base	*ARObject*, *FormulaExpression*, *SwSystemconstDependentFormula*			
Subclasses	*AbstractEnumerationValueVariationPoint*, *AbstractNumericalVariationPoint*, BooleanValueVariationPoint, FloatValueVariationPoint, IntegerValueVariationPoint, PositiveIntegerValueVariationPoint, TimeValue ValueVariationPoint, UnlimitedIntegerValueVariationPoint			
Aggregated by	VariationPointProxy.valueAccess			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
bindingTime	BindingTimeEnum	0..1	attr	This is the binding time in which the attribute value needs to be bound. If this attribute is missing, the attribute is not a variation point. In particular this means that It needs to be a single value according to the type specified in the pure model. It is an error if it is still a formula. **Tags:**xml.attribute=true
blueprintValue	String	0..1	attr	This represents a description that documents how the value shall be defined when deriving objects from the blueprint. **Tags:**xml.attribute=true
sd	String	0..1	attr	This special data is provided to allow synchronization of Attribute value variation points with variant management systems. The usage is subject of agreement between the involved parties. **Tags:**xml.attribute=true
shortLabel	PrimitiveIdentifier	0..1	attr	This allows to identify the variation point. It is also intended to allow RTE support for CompileTime Variation points. **Tags:**xml.attribute=true

Table A.67: AttributeValueVariationPoint

Class	**AutosarDataPrototype** (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::DataPrototypes			
Note	Base class for prototypical roles of an AutosarDataType.			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *DataPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	ArgumentDataPrototype, ParameterDataPrototype, VariableDataPrototype			
Aggregated by	*AtpClassifier*.atpFeature			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
type	AutosarDataType	0..1	tref	This represents the corresponding data type. **Stereotypes:** isOfType

Table A.68: AutosarDataPrototype

Class	**AutosarDataType** (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::Datatypes			
Note	Abstract base class for user defined AUTOSAR data types for software.			
Base	*ARElement*, *ARObject*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Subclasses	*AbstractImplementationDataType*, *ApplicationDataType*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
swDataDef Props	SwDataDefProps	0..1	aggr	The properties of this AutosarDataType. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=swDataDefProps

Table A.69: AutosarDataType

Class	**AutosarOperationArgumentInstance**			
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescription Events::TDEventOccurrenceExpression::InstanceRefsUsage			
Note	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: • 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 Client ServerInterface			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	TDEventOccurrenceExpression.argument, TimingExtensionResource.timingArgument			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
operation Argument Instance	DataPrototype	0..1	iref	This is the reference to the instanceRef definition. **InstanceRef implemented by:**OperationArgumentIn ComponentInstanceRef

Table A.70: AutosarOperationArgumentInstance

Class	AutosarParameterRef
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::DataElements
Note	This class represents a reference to a parameter within AUTOSAR which can be one of the following use cases: localParameter: • localParameter which is used as whole (e.g. sharedAxis for curve) autosarVariable: • a parameter provided via PortPrototype which is used as whole (e.g. parameterAccess) • an element inside of a composite local parameter typed by ApplicationDatatype (e.g. sharedAxis for a curve) • an element inside of a composite parameter provided via Port and typed by ApplicationDatatype (e.g. sharedAxis for a curve) autosarParameterInImplDatatype: • an element inside of a composite local parameter typed by ImplementationDatatype • an element inside of a composite parameter provided via PortPrototype and typed by ImplementationDatatype
Base	ARObject
Aggregated by	InstantiationDataDefProps.parameterInstance, ParameterAccess.accessedParameter, RoleBasedDataAssignment.usedParameterElement, SwCalprmRefProxy.arParameter

Attribute	Type	Mult.	Kind	Note
autosar Parameter	DataPrototype	0..1	iref	This instance reference is used if the calibration parameter is either imported via a port or is part of a composite data structure. **InstanceRef implemented by:**ParameterInAtomicSWCTypeInstanceRef
localParameter	DataPrototype	0..1	ref	In the majority of cases this reference goes to Parameter DataPrototypes rather than VariableDataPrototypes. Pointing the reference to a VariableDataPrototype is limited to special use cases, e.g. if the AutosarParameter Ref is used in the context of an SwAxisGrouped. This reference is used if the arParameter is local to the current component. Of course, it would technically also be feasible to use an InstanceRef for this case. However, the InstanceRef would not have a contextElement (because the current instance is the context). Hence, the local instance is a special case which may provide further optimization. Therefore an explicit reference is provided for this case.

Table A.71: AutosarParameterRef

Class	AutosarVariableInstance
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescription Events::TDEventOccurrenceExpression::InstanceRefsUsage
Note	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: • 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

▽

△

Class	AutosarVariableInstance			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
variableInstance	DataPrototype	0..1	iref	This is the reference to the instanceRef definition. **InstanceRef implemented by:**VariableInComponent InstanceRef

Table A.72: AutosarVariableInstance

Class	AutosarVariableRef			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::DataElements			
Note	This class represents a reference to a variable within AUTOSAR which can be one of the following use cases: localVariable: • localVariable which is used as whole (e.g. InterRunnableVariable, inputValue for curve) autosarVariable: • a variable provided via Port which is used as whole (e.g. dataAccesspoints) • an element inside of a composite local variable typed by ApplicationDatatype (e.g. inputValue for a curve) • an element inside of a composite variable provided via Port and typed by ApplicationDatatype (e.g. inputValue for a curve) autosarVariableInImplDatatype: • an element inside of a composite local variable typed by ImplementationDatatype (e.g. nvram Data mapping) • an element inside of a composite variable provided via Port and typed by Implementation Datatype (e.g. inputValue for a curve)			
Base	ARObject			
Aggregated by	InstantiationDataDefProps.variableInstance, NvBlockDataMapping.nvRamBlockElement, NvBlockData Mapping.readNvData, NvBlockDataMapping.writtenNvData, NvBlockDataMapping.writtenReadNvData, RoleBasedDataAssignment.usedDataElement, SwVariableRefProxy.autosarVariable, VariableAccess. accessedVariable			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
autosarVariable	DataPrototype	0..1	iref	This references a variable which is provided by a port and/or which is part of a CompositeDataType. **InstanceRef implemented by:**VariableInAtomicSWC TypeInstanceRef
autosarVariable InImplDatatype	ArVariableIn ImplementationData InstanceRef	0..1	aggr	This is used if the target variable is inside of variableData Prototype typed by an ImplementationDataType.
localVariable	VariableDataPrototype	0..1	ref	This reference is used if the variable is local to the current component. It would also be possible to use the instance refence here. Such an instance ref would not have a contextElement, since the current instance is the context. But the local instance is a special case which may provide further optimization. Therefore an explicit reference is provided for this case.

Table A.73: AutosarVariableRef

Class	**BaseType** (abstract)			
Package	M2::MSR::AsamHdo::BaseTypes			
Note	This abstract meta-class represents the ability to specify a platform dependent base type.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Subclasses	SwBaseType			
Aggregated by	ARPackage.element			
Attribute	**Type**	*Mult.*	*Kind*	**Note**
baseType Definition	BaseTypeDefinition	1	aggr	This is the actual definition of the base type. **Tags:** xml.roleElement=false xml.roleWrapperElement=false xml.sequenceOffset=20 xml.typeElement=false xml.typeWrapperElement=false

Table A.74: BaseType

Class	**BaseTypeDirectDefinition**			
Package	M2::MSR::AsamHdo::BaseTypes			
Note	This BaseType is defined directly (as opposite to a derived BaseType)			
Base	*ARObject*, *BaseTypeDefinition*			
Aggregated by	*BaseType*.baseTypeDefinition			
Attribute	**Type**	*Mult.*	*Kind*	**Note**
baseType Encoding	BaseTypeEncoding String	0..1	attr	This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **Tags:**xml.sequenceOffset=90
baseTypeSize	PositiveInteger	0..1	attr	Describes the length of the data type specified in the container in bits. **Tags:**xml.sequenceOffset=70
byteOrder	ByteOrderEnum	0..1	attr	This attribute specifies the byte order of the base type. **Tags:**xml.sequenceOffset=110
memAlignment	PositiveInteger	0..1	attr	This attribute describes the alignment of the memory object in bits. E.g. "8" specifies, that the object in question is aligned to a byte while "32" specifies that it is aligned four byte. If the value is set to "0" the meaning shall be interpreted as "unspecified". **Tags:**xml.sequenceOffset=100
native Declaration	NativeDeclarationString	0..1	attr	This attribute describes the declaration of such a base type in the native programming language, primarily in the Programming language C. This can then be used by a code generator to include the necessary declarations into a header file. For example BaseType with shortName: "MyUnsignedInt" native Declaration: "unsigned short" Results in typedef unsigned short MyUnsignedInt; If the attribute is not defined the referring Implementation DataTypes will not be generated as a typedef by RTE. If a nativeDeclaration type is given it shall fulfill the characteristic given by basetypeEncoding and baseType Size.

▽

△

Class	BaseTypeDirectDefinition			
				△ This is required to ensure the consistent handling and interpretation by software components, RTE, COM and MCM systems. **Tags:**xml.sequenceOffset=120

Table A.75: BaseTypeDirectDefinition

Class	Baseline			
Package	M2::AUTOSARTemplates::CommonStructure::StandardizationTemplate::DataExchangePoint			
Note	Specification of the baseline of the AUTOSAR standard this Data Exchange Point relates to. The baseline is specified by listing the AUTOSAR products and their revisions. Custom defined functionality and deviations to the standard can be provided as well. All references to specification elements in this Data Exchange Point refer to specification elements that are part of this specification baseline.			
Base	ARObject			
Aggregated by	DataExchangePoint.referencedBaseline			
Attribute	Type	Mult.	Kind	Note
customSdgDef	SdgDef	*	ref	Reference to custom SdgDefs that extend the data format of this baseline, **Tags:**xml.sequenceOffset=30
custom Specification	Documentation	*	ref	Reference to custom specifications that extend this baseline, **Tags:**xml.sequenceOffset=20
standard Revision	String	*	attr	Specifies a combination of revisions of AUTOSAR standards that are used as the specification baseline of this Data Exchange Point. All standard specification elements that are referenced by this Profile of Data Exchange Point have to be part of specifications that belong to the defined AUTOSAR standards. **Tags:**xml.sequenceOffset=10

Table A.76: Baseline

Class	BinaryManifestAddressableObject (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::SoftwareCluster::BinaryManifest			
Note	This meta-class acts as an abstract base class for addressable objects in the context of the binary manifest of a CP software cluster. **Tags:**atp.Status=draft			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Subclasses	BinaryManifestItem, BinaryManifestMetaDataField			
Attribute	Type	Mult.	Kind	Note
address	Address	0..1	attr	This attribute specifies the address of the enclosing addressable object. **Tags:**atp.Status=draft
symbol	SymbolString	0..1	attr	This attribute specifies the symbol of the addressable object. **Tags:**atp.Status=draft

Table A.77: BinaryManifestAddressableObject

Class	BinaryManifestItem			
Package	M2::AUTOSARTemplates::SystemTemplate::SoftwareCluster::BinaryManifest			
Note	This meta-class represents the ability to describe a specific handle or auxiliary field in the context of binary manifest resource. **Tags:**atp.Status=draft			
Base	*ARObject*, *BinaryManifestAddressableObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	BinaryManifestItem.auxiliaryField, *BinaryManifestResource*.item			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
auxiliaryField	BinaryManifestItem	*	aggr	This aggregation is used to define structured Binary ManifestItems. **Tags:** atp.Status=draft xml.sequenceOffset=20
defaultValue	BinaryManifestItem Value	0..1	aggr	This aggregation represents the definition of a default value for a binary manifest handle or an auxiliaryField. This value shall be taken if no connection for this resource is possible. **Tags:** atp.Status=draft xml.sequenceOffset=10
isUnused	Boolean	0..1	attr	If true, the handle or auxiliary field in the context of binary manifest resource relates to an optional BinaryManifest ItemDefinition and is not used. **Tags:**atp.Status=draft
value	BinaryManifestItem Value	0..1	aggr	This aggregation represents the definition of a value for a binary manifest handle or an auxiliaryField. This value shall be taken to establish a connection. **Tags:**atp.Status=draft

Table A.78: BinaryManifestItem

Class	BinaryManifestItemDefinition			
Package	M2::AUTOSARTemplates::SystemTemplate::SoftwareCluster::BinaryManifest			
Note	This meta-class provides the ability to define the handle definition or an auxiliary field of a binary manifest resource. **Tags:**atp.Status=draft			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	BinaryManifestItemDefinition.auxiliaryFieldDefinition, BinaryManifestResourceDefinition.itemDefinition			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
auxiliaryField Definition	BinaryManifestItem Definition	*	aggr	This aggregation is used to define structured Binary ManifestItemDefinitions. **Tags:**atp.Status=draft
isOptional	Boolean	0..1	attr	If true, the handle definition or auxiliary field of a binary manifest resource is optional and may not be used in all BinaryManifestResources referring to this BinaryManifest ResourceDefinition. **Tags:**atp.Status=draft
size	PositiveInteger	0..1	attr	This attribute provides the ability to specify the size of the enclosing BinaryManifestResourceDefinition. **Tags:**atp.Status=draft

Table A.79: BinaryManifestItemDefinition

Class	BinaryManifestItemNumericalValue			
Package	M2::AUTOSARTemplates::SystemTemplate::SoftwareCluster::BinaryManifest			
Note	This meta-class has the ability to provide a numerical value for a binary manifest item. **Tags:**atp.Status=draft			
Base	*ARObject*, *BinaryManifestItemValue*			
Aggregated by	BinaryManifestItem.defaultValue, BinaryManifestItem.value			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
value	Numerical	0..1	attr	This attribute specifies the actual numerical value to be used in the binary manifest handle. **Tags:**atp.Status=draft

Table A.80: BinaryManifestItemNumericalValue

Class	BinaryManifestItemPointerValue			
Package	M2::AUTOSARTemplates::SystemTemplate::SoftwareCluster::BinaryManifest			
Note	This meta-class has the ability to provide a value for a pointer in the context of a binary manifest item. **Tags:**atp.Status=draft			
Base	*ARObject*, *BinaryManifestItemValue*			
Aggregated by	BinaryManifestItem.defaultValue, BinaryManifestItem.value			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
address	Address	0..1	attr	This attribute represents the address value of the enclosing pointer value. **Tags:**atp.Status=draft
symbol	SymbolString	0..1	attr	This attribute represents the symbol associated with the binary manifest handle. **Tags:**atp.Status=draft

Table A.81: BinaryManifestItemPointerValue

Class	BinaryManifestMetaDataField			
Package	M2::AUTOSARTemplates::SystemTemplate::SoftwareCluster::BinaryManifest			
Note	This meta-class provides the ability to define a meta-data field for the binary manifest descriptor. **Tags:**atp.Status=draft			
Base	*ARObject*, *BinaryManifestAddressableObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CpSoftwareClusterBinaryManifestDescriptor.metaDataField			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
size	PositiveInteger	0..1	attr	The value of this attribute represents the size of the meta-data field in bytes. **Tags:**atp.Status=draft
value	VerbatimString	0..1	attr	This attribute specifies the value of the meta-data field. **Tags:**atp.Status=draft

Table A.82: BinaryManifestMetaDataField

Class	BinaryManifestProvideResource			
Package	M2::AUTOSARTemplates::SystemTemplate::SoftwareCluster::BinaryManifest			
Note	This meta-class represents a provided resource in the binary manifest. **Tags:**atp.Status=draft			
Base	*ARObject*, *BinaryManifestResource*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CpSoftwareClusterBinaryManifestDescriptor.provideResource			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
numberOf NotifierSets	PositiveInteger	0..1	attr	This attribute provides an upper limit for the number of notifiers for this resource. **Tags:**atp.Status=draft
supports MultipleNotifier Sets	Boolean	0..1	attr	This attribute indicates whether the enclosing Binary ManifestResource supports multiple notifiers sets. **Tags:**atp.Status=draft

Table A.83: BinaryManifestProvideResource

Class	BinaryManifestRequireResource			
Package	M2::AUTOSARTemplates::SystemTemplate::SoftwareCluster::BinaryManifest			
Note	This meta-class represents a required resource in the binary manifest. **Tags:**atp.Status=draft			
Base	*ARObject*, *BinaryManifestResource*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CpSoftwareClusterBinaryManifestDescriptor.requireResource			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
connectionIs Mandatory	Boolean	0..1	attr	This attribute indicates whether the connection of the enclosing BinaryManifestResource is mandatory. **Tags:**atp.Status=draft

Table A.84: BinaryManifestRequireResource

Class	BinaryManifestResource (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::SoftwareCluster::BinaryManifest			
Note	This meta-class acts as an abstract base class for specializations. **Tags:**atp.Status=draft			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	BinaryManifestProvideResource, BinaryManifestRequireResource			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
globalResource Id	PositiveInteger	0..1	attr	A unique identifiers per resource used for the connection process. The identifier is required to be unique in the scope of a single machine. If software clusters are designed to be reused on multiple machines the uniqueness requirements applies for all the intended machines. **Tags:**atp.Status=draft
item (ordered)	BinaryManifestItem	*	aggr	This aggregation represents the collection of binary manifest handles owned by the enclosing binary manifest resource. **Tags:**atp.Status=draft

▽

△

Class	BinaryManifestResource (abstract)			
resource	CpSoftwareCluster Resource	0..1	ref	This reference identifies the CpSoftwareClusterResource (on design level) that corresponds to the BinaryManifest Resource (on integration level). **Tags:**atp.Status=draft
resource Definition	BinaryManifest ResourceDefinition	0..1	ref	this reference identifies the definition of the Binary ManifestResource. The definition provides configuration information that is shared among all BinaryManifest Resources that refer to the BinaryManifestResource Definition. **Tags:**atp.Status=draft
resourceGuard Value	String	0..1	attr	This attribute specifies the guard value of the enclosing binary manifest resource. **Tags:**atp.Status=draft

Table A.85: BinaryManifestResource

Class	BinaryManifestResourceDefinition			
Package	M2::AUTOSARTemplates::SystemTemplate::SoftwareCluster::BinaryManifest			
Note	This meta-class represents the ability to specify a resource definition that provides information that can be shared by all resources that refer to the respective resource definition. **Tags:**atp.Status=draft			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	CpSoftwareClusterBinaryManifestDescriptor.resourceDefinition			
Attribute	Type	Mult.	Kind	Note
itemDefinition (ordered)	BinaryManifestItem Definition	*	aggr	This aggregation specifies the collection of handle definitions in the context of the enclosing binary manifest resource definitions. **Tags:**atp.Status=draft

Table A.86: BinaryManifestResourceDefinition

Class	BlueprintMapping			
Package	M2::AUTOSARTemplates::CommonStructure::StandardizationTemplate::BlueprintDedicated::Generic Blueprint			
Note	This meta-class represents the ability to map two an object and its blueprint.			
Base	ARObject, AtpBlueprintMapping			
Aggregated by	BlueprintMappingSet.blueprintMap			
Attribute	Type	Mult.	Kind	Note
blueprint	AtpBlueprint	1	ref	This represents the mapped blueprint.
derivedObject	AtpBlueprintable	1	ref	This represents the object which was derived from the blueprint.

Table A.87: BlueprintMapping

Class	BlueprintPolicy (abstract)			
Package	M2::AUTOSARTemplates::CommonStructure::StandardizationTemplate::AbstractBlueprintStructure			
Note	This meta-class represents the ability to indicate whether blueprintable elements will be modifiable or not modifiable.			
Base	ARObject			
Subclasses	BlueprintPolicyModifiable, BlueprintPolicyNotModifiable			
Aggregated by	AtpBlueprint.blueprintPolicy			
Attribute	Type	Mult.	Kind	Note
attributeName	String	1	attr	This identifies the related attribute of a BlueprintPolicy. For navigation over the model a subset of xpath expressions is used.

Table A.88: BlueprintPolicy

Class	BlueprintPolicyNotModifiable			
Package	M2::AUTOSARTemplates::CommonStructure::StandardizationTemplate::AbstractBlueprintStructure			
Note	The class represents that the related attribute is not modifiable during the blueprinting.			
Base	ARObject, BlueprintPolicy			
Aggregated by	AtpBlueprint.blueprintPolicy			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.89: BlueprintPolicyNotModifiable

Class	BswAsynchronousServerCallPoint			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	Represents an asynchronous procedure call point via the BSW Scheduler.			
Base	ARObject, BswModuleCallPoint, Referrable			
Aggregated by	BswModuleEntity.callPoint			
Attribute	Type	Mult.	Kind	Note
calledEntry	BswModuleClientServer Entry	0..1	ref	The entry to be called.

Table A.90: BswAsynchronousServerCallPoint

Class	BswAsynchronousServerCallResultPoint			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	The callback point for an BswAsynchronousServerCallPoint i.e. the point at which the result can be retrieved from the BSW Scheduler.			
Base	ARObject, BswModuleCallPoint, Referrable			
Aggregated by	BswModuleEntity.callPoint			
Attribute	Type	Mult.	Kind	Note
asynchronous ServerCallPoint	BswAsynchronous ServerCallPoint	0..1	ref	The call point invoking the call to which the result belongs.

Table A.91: BswAsynchronousServerCallResultPoint

Class	BswAsynchronousServerCallReturnsEvent			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	This is the "callback" event for asynchronous Client-Server-Communication via the BSW Scheduler which is thrown after completion of the asynchronous Client-Server call. Its eventSource specifies the call point to be used for retrieving the result.			
Base	ARObject, AbstractEvent, BswEvent, BswScheduleEvent, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	BswInternalBehavior.event			
Attribute	Type	Mult.	Kind	Note
eventSource	BswAsynchronous ServerCallResultPoint	0..1	ref	The call point to be used for retrieving the result.

Table A.92: BswAsynchronousServerCallReturnsEvent

Enumeration	BswCallType
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswInterfaces
Note	Denotes the mechanism by which the entry into the Bsw module shall be called.
Aggregated by	BswModuleEntry.callType
Literal	Description
callback	Callback (i.e. the caller specifies the signature) **Tags:**atp.EnumerationLiteralIndex=0
callout	Callout - provide defined means to extend the functionality of an existing module. In this case caller specifies the signature. **Tags:**atp.EnumerationLiteralIndex=4
interrupt	Interrupt routine **Tags:**atp.EnumerationLiteralIndex=1
regular	Regular API call **Tags:**atp.EnumerationLiteralIndex=2
scheduled	Called by the scheduler **Tags:**atp.EnumerationLiteralIndex=3

Table A.93: BswCallType

Class	BswCalledEntity			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	BSW module entity which is designed to be called from another BSW module or cluster.			
Base	ARObject, BswModuleEntity, ExecutableEntity, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	BswInternalBehavior.entity			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.94: BswCalledEntity

Class	BswCompositionTiming			
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingExtensions			
Note	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. TimingDescriptions aggregated by BswCompositionTiming are restricted to event chains referring to events which are derived from the class TDEventBswInternalBehavior and TDEventBsw. **Tags:**atp.recommendedPackage=TimingExtensions			
Base	*ARElement*, *ARObject*, *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 A.95: BswCompositionTiming

Class	BswDataReceivedEvent			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	This event is thrown on reception of the referenced data via Sender-Receiver-Communication over the BSW Scheduler.			
Base	*ARObject*, *AbstractEvent*, *BswEvent*, *BswScheduleEvent*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	BswInternalBehavior.event			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
data	VariableDataPrototype	0..1	ref	The received data.

Table A.96: BswDataReceivedEvent

Class	*BswDataReceptionPolicy* (abstract)			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	Specifies the reception policy for the referred data in sender-receiver communication over the BSW Scheduler. To be used for inter-partition and/or inter-core communication.			
Base	*ARObject*, *BswApiOptions*			
Subclasses	BswQueuedDataReceptionPolicy			
Aggregated by	BswInternalBehavior.receptionPolicy			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
receivedData	VariableDataPrototype	0..1	ref	The data received over the BSW Scheduler using this policy.

Table A.97: BswDataReceptionPolicy

Class	BswDirectCallPoint			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	Represents a concrete point in the code from where a BswModuleEntry is called directly, i.e. not via the BSW Scheduler. This information can be used to analyze call tree and resource locking scenarios. It is not needed to configure the BSW Scheduler.			

▽

△

Class	BswDirectCallPoint			
Base	ARObject, *BswModuleCallPoint*, *Referrable*			
Aggregated by	*BswModuleEntity*.callPoint			
Attribute	Type	Mult.	Kind	Note
calledEntry	BswModuleEntry	0..1	ref	The BswModuleEntry called at this point.
calledFrom WithinExclusive Area	ExclusiveAreaNesting Order	0..1	ref	This indicates that the call point is located at the deepest level inside one or more ExclusiveAreas that are nested in the given order.

Table A.98: BswDirectCallPoint

Class	BswDistinguishedPartition			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	Each instance of this meta-class represents an abstract partition in which context the code of the enclosing BswModuleBehavior can be executed. The intended use case is to distinguish between several partitions in order to implement different behavior per partition, for example to behave either as a master or satellite in a multicore ECU with shared BSW code.			
Base	ARObject, *Referrable*			
Aggregated by	BswInternalBehavior.distinguishedPartition			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.99: BswDistinguishedPartition

Class	BswEntryRelationship			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswInterfaces			
Note	Describes a relationship between two BswModuleEntrys and the type of relationship.			
Base	ARObject			
Aggregated by	BswEntryRelationshipSet.bswEntryRelationship			
Attribute	Type	Mult.	Kind	Note
bswEntry Relationship Type	BswEntryRelationship Enum	0..1	attr	Denotes the type of the relationship. **Tags:**xml.sequenceOffset=5
from	BswModuleEntry	0..1	ref	Type of relationship that refers to the abstract BswModule Entry. Please notice that in this case the bswEntry RelationshipType shall be set to drivedFrom.
to	BswModuleEntry	0..1	ref	Type of relationship that refers to the concrete Bsw ModuleEntry

Table A.100: BswEntryRelationship

Class	BswEntryRelationshipSet			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswInterfaces			
Note	Describes a set of relationships between two BswModuleEntrys. **Tags:**atp.recommendedPackage=BswEntryRelationshipSets			
Base	*ARElement*, ARObject, *AtpBlueprint*, AtpBlueprintable, CollectableElement, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			

▽

△

Class	BswEntryRelationshipSet			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
bswEntry Relationship	BswEntryRelationship	*	aggr	Relationship between two BswModuleEntrys.

Table A.101: BswEntryRelationshipSet

Class	*BswEvent* (abstract)			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	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.			
Base	*ARObject*, *AbstractEvent*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	BswOperationInvokedEvent, *BswScheduleEvent*			
Aggregated by	BswInternalBehavior.event			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
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	The modes, in which this event is disabled. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=disabledInMode.contextMode DeclarationGroup, disabledInMode.targetMode **InstanceRef implemented by:**ModeInBswModule DescriptionInstanceRef
startsOnEvent	BswModuleEntity	0..1	ref	The entity which is started by the event.

Table A.102: BswEvent

Class	BswExclusiveAreaPolicy			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	The ExclusiveArea for which the BSW Scheduler using this policy.			
Base	*ARObject*, *BswApiOptions*			
Aggregated by	BswInternalBehavior.exclusiveAreaPolicy			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
apiPrinciple	ApiPrincipleEnum	0..1	attr	Specifies for this ExclusiveArea if either one common set of Enter and Exit APIs for the whole BSW module is requested from the SchM or if the set of Enter and Exit APIs is expected per BswModuleEntity. The default value is "common".
exclusiveArea	ExclusiveArea	0..1	ref	The ExclusiveArea for which the BSW Scheduler using this policy.

Table A.103: BswExclusiveAreaPolicy

Enumeration	BswExecutionContext
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswInterfaces
Note	Specifies the execution context required or guaranteed for the call associated with this service.
Aggregated by	BswModuleEntry.executionContext
Literal	Description
hook	Context of an OS "hook" routine always **Tags:**atp.EnumerationLiteralIndex=0
interruptCat1	CAT1 interrupt context always **Tags:**atp.EnumerationLiteralIndex=1
interruptCat2	CAT2 interrupt context always **Tags:**atp.EnumerationLiteralIndex=2
task	Task context always **Tags:**atp.EnumerationLiteralIndex=3
unspecified	The execution context is not specified by the API **Tags:**atp.EnumerationLiteralIndex=4

Table A.104: BswExecutionContext

Class	BswExternalTriggerOccurredEvent			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	A BswEvent resulting from a trigger released by another module or cluster.			
Base	ARObject, AbstractEvent, BswEvent, BswScheduleEvent, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	BswInternalBehavior.event			
Attribute	Type	Mult.	Kind	Note
trigger	Trigger	0..1	ref	The trigger associated with this event. The trigger is external to this module.

Table A.105: BswExternalTriggerOccurredEvent

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	Version of the AUTOSAR Release on which this implementation is based. The numbering contains three levels (major, minor, revision) which are defined by AUTOSAR.

▽

△

Class	BswImplementation			
behavior	BswInternalBehavior	0..1	ref	The behavior of this implementation. This relation is made as an association because • it follows the pattern of the SWCT • since ARElement cannot be splitted, but we want supply the implementation later, the Bsw Implementation is not aggregated in BswBehavior
preconfigured Configuration	EcucModule ConfigurationValues	*	ref	Reference to the set of preconfigured (i.e. fixed) configuration values for this BswImplementation. 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. **Tags:**xml.roleWrapperElement=true
recommended Configuration	EcucModule ConfigurationValues	*	ref	Reference to one or more sets of recommended configuration values for this module or module cluster.
vendorApiInfix	Identifier	0..1	attr	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>. 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. This attribute is mandatory for all modules with upper multiplicity > 1. It shall not be used for modules with upper multiplicity =1. See also SWS_BSW_00102.
vendorSpecific ModuleDef	EcucModuleDef	*	ref	Reference to • the vendor specific EcucModuleDef used in this BswImplementation 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 **Tags:**xml.roleWrapperElement=true

Table A.106: 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*, *Identifiable*, *InternalBehavior*, *Multilanguage Referrable*, *Referrable*

▽

△

Class	BswInternalBehavior			
Aggregated by	*AtpClassifier*.atpFeature, BswModuleDescription.internalBehavior			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
arTypedPer Instance Memory	VariableDataPrototype	*	aggr	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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=arTypedPerInstanceMemory.shortName, ar TypedPerInstanceMemory.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
bswPerInstance MemoryPolicy	BswPerInstance MemoryPolicy	*	aggr	Policy for a arTypedPerInstanceMemory The policy selects the options of the Schedule Manager API generation. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=bswPerInstanceMemoryPolicy, bswPer InstanceMemoryPolicy.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
clientPolicy	BswClientPolicy	*	aggr	Policy for a requiredClientServerEntry. The policy selects the options of the Schedule Manager API generation. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=clientPolicy, clientPolicy.variationPoint.short Label vh.latestBindingTime=preCompileTime
distinguished Partition	BswDistinguished Partition	*	aggr	Indicates an abstract partition context in which the enclosing BswModuleEntity can be executed. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=distinguishedPartition.shortName, distinguishedPartition.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=60
entity	BswModuleEntity	*	aggr	A code entity for which the behavior is described **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=entity.shortName, entity.variationPoint.short Label vh.latestBindingTime=preCompileTime xml.sequenceOffset=5
event	BswEvent	*	aggr	An event required by this module behavior. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=event.shortName, event.variationPoint.short Label vh.latestBindingTime=preCompileTime xml.sequenceOffset=10

▽

△

Class	**BswInternalBehavior**			
exclusiveArea Policy	BswExclusiveArea Policy	*	aggr	Policy for an ExclusiveArea in this BswInternalBehavior. The policy selects the options of the Schedule Manager API generation. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=exclusiveAreaPolicy, exclusiveArea Policy.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
includedData TypeSet	IncludedDataTypeSet	*	aggr	The includedDataTypeSet is used by a basic software module for its implementation. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=includedDataTypeSet
includedMode Declaration GroupSet	IncludedMode DeclarationGroupSet	*	aggr	This aggregation represents the included Mode DeclarationGroups **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=includedModeDeclarationGroupSet
internal TriggeringPoint	BswInternalTriggering Point	*	aggr	An internal triggering point. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=internalTriggeringPoint.shortName, internal TriggeringPoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=2
internal TriggeringPoint Policy	BswInternalTriggering PointPolicy	*	aggr	Policy for an internalTriggeringPoint in this BswInternal Behavior.. The policy selects the options of the Schedule Manager API generation. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=internalTriggeringPointPolicy, internal TriggeringPointPolicy.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
modeReceiver Policy	BswModeReceiver Policy	*	aggr	Implementation policy for the reception of mode switches. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=modeReceiverPolicy, modeReceiver Policy.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=25
modeSender Policy	BswModeSenderPolicy	*	aggr	Implementation policy for providing a mode group. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=modeSenderPolicy, modeSender Policy.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=20
parameterPolicy	BswParameterPolicy	*	aggr	Policy for a perInstanceParameter in this BswInternal Behavior. The policy selects the options of the Schedule Manager API generation. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=parameterPolicy, parameterPolicy.variation Point.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	**BswInternalBehavior**			
perInstance Parameter	ParameterData Prototype	*	aggr	Describes a read only memory object containing characteristic value(s) needed by this BswInternal Behavior. The role name perInstanceParameter is chosen in analogy to the similar role in the context of SwcInternal Behavior. 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. The aggregation is subject to variability with the purpose to support implementation variants. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=perInstanceParameter.shortName, per InstanceParameter.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=45
receptionPolicy	BswDataReception Policy	*	aggr	Data reception policy for inter-partition and/or inter-core communication. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=receptionPolicy, receptionPolicy.variation Point.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=55
releasedTrigger Policy	BswReleasedTrigger Policy	*	aggr	Policy for a releasedTrigger. The policy selects the options of the Schedule Manager API generation. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=releasedTriggerPolicy, releasedTrigger Policy.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
schedulerName Prefix	BswSchedulerName Prefix	*	aggr	Optional definition of one or more prefixes to be used for the BswScheduler. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=schedulerNamePrefix.shortName, scheduler NamePrefix.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=50
sendPolicy	BswDataSendPolicy	*	aggr	Policy for a providedData. The policy selects the options of the Schedule Manager API generation. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=sendPolicy, sendPolicy.variationPoint.short Label vh.latestBindingTime=preCompileTime

▽

△

Class	BswInternalBehavior			
service Dependency	BswService Dependency	*	aggr	Defines the requirements on AUTOSAR Services for a particular item.
				The aggregation is subject to variability with the purpose to support the conditional existence of ServiceNeeds.
				The aggregation is splitable in order to support that ServiceNeeds might be provided in later development steps.
				Stereotypes: atpSplitable; atpVariation **Tags:** atp.Splitkey=serviceDependency.ident.shortName, serviceDependency.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=40
triggerDirect Implementation	BswTriggerDirect Implementation	*	aggr	Specifies a trigger to be directly implemented via OS calls.
				Stereotypes: atpSplitable; atpVariation **Tags:** atp.Splitkey=triggerDirectImplementation, triggerDirect Implementation.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=15
variationPoint Proxy	VariationPointProxy	*	aggr	Proxy of a variation points in the C/C++ implementation.
				Stereotypes: atpSplitable **Tags:**atp.Splitkey=variationPointProxy.shortName

Table A.107: BswInternalBehavior

Class	BswInternalTriggerOccurredEvent			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	A BswEvent, which can happen sporadically. The event is activated by explicit calls from the module to the BSW Scheduler. The main purpose for such an event is to cause a context switch, e.g. from an ISR context into a task context. Activation and switching are handled within the same module or cluster only.			
Base	ARObject, *AbstractEvent*, *BswEvent*, *BswScheduleEvent*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	BswInternalBehavior.event			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
eventSource	BswInternalTriggering Point	0..1	ref	The activation point is the source of this event.

Table A.108: BswInternalTriggerOccurredEvent

Class	BswInternalTriggeringPoint			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	Represents the activation point for one or more BswInternalTriggerOccurredEvents.			
Base	ARObject, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	BswInternalBehavior.internalTriggeringPoint			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
swImplPolicy	SwImplPolicyEnum	0..1	attr	This attribute, when set to value queued, specifies a queued processing of the internal trigger event.

Table A.109: BswInternalTriggeringPoint

Class	BswInterruptEntity			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	BSW module entity, which is designed to be triggered by an interrupt.			
Base	*ARObject*, *BswModuleEntity*, *ExecutableEntity*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	BswInternalBehavior.entity			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
interrupt Category	BswInterruptCategory	0..1	attr	Category of the interrupt
interruptSource	String	0..1	attr	Allows a textual documentation of the intended interrupt source.

Table A.110: BswInterruptEntity

Class	BswMgrNeeds			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	Specifies the abstract needs on the configuration of the Basic Software Manager for one "user".			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *ServiceNeeds*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.111: BswMgrNeeds

Class	BswModeManagerErrorEvent			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	This represents the ability to react on errors occurring during mode handling.			
Base	*ARObject*, *AbstractEvent*, *BswEvent*, *BswScheduleEvent*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	BswInternalBehavior.event			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
modeGroup	ModeDeclarationGroup Prototype	0..1	ref	This represents the ModeDeclarationGroupPrototype for which the error behavior of the mode manager applies.

Table A.112: BswModeManagerErrorEvent

Class	BswModeReceiverPolicy			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	Specifies the details for the reception of a mode switch for the referred mode group.			
Base	*ARObject*			
Aggregated by	BswInternalBehavior.modeReceiverPolicy			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
enhancedMode Api	Boolean	0..1	attr	This controls the creation of the enhanced mode API that returns information about the previous mode and the next mode. If set to TRUE the enhanced mode API is supposed to be generated. For more details please refer to the SWS_RTE.
requiredMode Group	ModeDeclarationGroup Prototype	0..1	ref	The required mode group for which the policy is specified.

▽

△

Class	BswModeReceiverPolicy			
supports Asynchronous ModeSwitch	Boolean	0..1	attr	Specifies whether the module can handle the reception of an asynchronous mode switch (true) or not (false).

Table A.113: BswModeReceiverPolicy

Class	BswModeSenderPolicy			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	Specifies the details for the sending of a mode switch for the referred mode group.			
Base	ARObject			
Aggregated by	BswInternalBehavior.modeSenderPolicy			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
ackRequest	BswModeSwitchAck Request	0..1	aggr	Request for acknowledgement
enhancedMode Api	Boolean	0..1	attr	This controls the creation of the enhanced mode API that returns information about the previous mode and the next mode. If set to TRUE the enhanced mode API is supposed to be generated. For more details please refer to the SWS_RTE.
providedMode Group	ModeDeclarationGroup Prototype	0..1	ref	The provided mode group for which the policy is specified.
queueLength	PositiveInteger	0..1	attr	Length of call queue on the sender side. The queue is implemented by the RTE resp.BswScheduler. The value shall be greater or equal to 0. Setting the value of queue Length to 0 implies non-queued communication.

Table A.114: BswModeSenderPolicy

Class	BswModeSwitchAckRequest			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	Requests acknowledgements that a mode switch has been processed successfully			
Base	ARObject			
Aggregated by	BswModeSenderPolicy.ackRequest			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
timeout	TimeValue	0..1	attr	Number of seconds before an error is reported.

Table A.115: BswModeSwitchAckRequest

Class	BswModeSwitchEvent			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	A BswEvent resulting from a mode switch.			
Base	ARObject, AbstractEvent, BswEvent, BswScheduleEvent, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	BswInternalBehavior.event			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
activation	ModeActivationKind	0..1	attr	Kind of activation w.r.t. to the referred mode.

▽

△

Class	BswModeSwitchEvent			
mode (ordered)	ModeDeclaration	0..2	iref	Reference to one or two Modes that initiate the Mode Switch Event. **InstanceRef implemented by:**ModeInBswModule DescriptionInstanceRef

Table A.116: BswModeSwitchEvent

Class	BswModeSwitchedAckEvent			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	The event is raised after a switch of the referenced mode group has been acknowledged or an error occurs. The referenced mode group shall be provided by this module.			
Base	*ARObject*, *AbstractEvent*, *BswEvent*, *BswScheduleEvent*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	BswInternalBehavior.event			
Attribute	Type	Mult.	Kind	Note
modeGroup	ModeDeclarationGroup Prototype	0..1	ref	A mode group provided by this module. The acknowledgement of a switch of this group raises this event.

Table A.117: BswModeSwitchedAckEvent

Class	BswModuleCallPoint (abstract)			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	Represents a point at which a BswModuleEntity handles a procedure call into a BswModuleEntry, either directly or via the BSW Scheduler.			
Base	*ARObject*, *Referrable*			
Subclasses	BswAsynchronousServerCallPoint, BswAsynchronousServerCallResultPoint, BswDirectCallPoint, Bsw SynchronousServerCallPoint			
Aggregated by	*BswModuleEntity*.callPoint			
Attribute	Type	Mult.	Kind	Note
context Limitation	BswDistinguished Partition	*	ref	The existence of this reference indicates that the call point is used only in the context of the referred Bsw DistinguishedPartitions.

Table A.118: BswModuleCallPoint

Class	BswModuleClientServerEntry			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswInterfaces			
Note	This meta-class represents a single API entry into the BSW module or cluster that has the ability to be called in client-server fashion via the BSW Scheduler. In this regard it is more special than BswModuleEntry and can be seen as a wrapper around the Bsw ModuleEntry to which it refers (property encapsulatedEntry). **Tags:**atp.recommendedPackage=BswModuleEntrys			
Base	*ARObject*, *Referrable*			
Aggregated by	BswModuleDescription.providedClientServerEntry, BswModuleDescription.requiredClientServerEntry			
Attribute	Type	Mult.	Kind	Note
encapsulated Entry	BswModuleEntry	0..1	ref	The underlying BswModuleEntry. **Tags:**xml.sequenceOffset=5

▽

$\triangle$

Class	BswModuleClientServerEntry			
isReentrant	Boolean	0..1	attr	Reentrancy from the viewpoint of clients invoking the service via the BSW Scheduler: • True: Enables the service to be invoked again, before the service has finished. • False: It is prohibited to invoke the service again before is has finished. **Tags:**xml.sequenceOffset=10
isSynchronous	Boolean	0..1	attr	Synchronicity from the viewpoint of clients invoking the service via the BSW Scheduler: • 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. **Tags:**xml.sequenceOffset=15

Table A.119: BswModuleClientServerEntry

Class	BswModuleDependency			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswInterfaces			
Note	This class collects the dependencies of a BSW module or cluster on a certain other BSW module.			
Base	ARObject, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	BswModuleDescription.bswModuleDependency			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
targetModuleId	PositiveInteger	0..1	attr	AUTOSAR identifier of the target module of which the dependencies are defined. This information is optional, because the target module may also be identified by targetModuleRef. **Tags:**xml.sequenceOffset=5
targetModule Ref	BswModuleDescription	0..1	ref	Reference to the target module. It is an <<atpUriDef>> because the reference shall be used to identify the target module without actually needing the description of that target module. **Stereotypes:** atpSplitable; atpUriDef; atpVariation **Tags:** atp.Splitkey=targetModuleRef.bswModuleDescription, targetModuleRef.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=7

Table A.120: BswModuleDependency

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	*ARElement*, ARObject, *AtpBlueprint*, AtpBlueprintable, *AtpClassifier*, AtpFeature, *AtpStructureElement*, CollectableElement, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*
Aggregated by	ARPackage.element, *AtpClassifier*.atpFeature

$\triangledown$

△

Class	**BswModuleDescription**			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
bswModule Dependency	BswModuleDependency	*	aggr	Describes the dependency to another BSW module. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=bswModuleDependency.shortName, bsw ModuleDependency.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=20
bswModule Documentation	SwComponent Documentation	0..1	aggr	This adds a documentation to the BSW module. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=bswModuleDocumentation, bswModule Documentation.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=6
expectedEntry	BswModuleEntry	*	ref	Indicates an entry which is required by this module. Replacement of outgoingCallback / requiredEntry. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=expectedEntry.bswModuleEntry, expected Entry.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
implemented Entry	BswModuleEntry	*	ref	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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=implementedEntry.bswModuleEntry, implementedEntry.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
internalBehavior	BswInternalBehavior	*	aggr	The various BswInternalBehaviors associated with a Bsw ModuleDescription can be distributed over several physical files. Therefore the aggregation is <<atp Splitable>>. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=internalBehavior.shortName xml.sequenceOffset=65
moduleId	PositiveInteger	0..1	attr	Refers to the BSW Module Identifier defined by the AUTOSAR standard. For non-standardized modules, a proprietary identifier can be optionally chosen. **Tags:**xml.sequenceOffset=5
providedClient ServerEntry	BswModuleClientServer Entry	*	aggr	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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=providedClientServerEntry.shortName, providedClientServerEntry.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=45

▽

$\triangle$

Class	**BswModuleDescription**			
providedData	VariableDataPrototype	*	aggr	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.
				Stereotypes: atpSplitable; atpVariation **Tags:** atp.Splitkey=providedData.shortName, provided Data.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=55
providedMode Group	ModeDeclarationGroup Prototype	*	aggr	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, EcuAbstraction SwComponentType or ComplexDeviceDriverSw ComponentType.
				Stereotypes: atpSplitable; atpVariation **Tags:** atp.Splitkey=providedModeGroup.shortName, provided ModeGroup.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=25
releasedTrigger	Trigger	*	aggr	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.
				Stereotypes: atpSplitable; atpVariation **Tags:** atp.Splitkey=releasedTrigger.shortName, released Trigger.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=35
requiredClient ServerEntry	BswModuleClientServer Entry	*	aggr	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.
				Stereotypes: atpSplitable; atpVariation **Tags:** atp.Splitkey=requiredClientServerEntry.shortName, requiredClientServerEntry.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=50

$\triangledown$

△

Class	BswModuleDescription			
requiredData	VariableDataPrototype	*	aggr	Specifies a data prototype required by this module in oder 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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=requiredData.shortName, required Data.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=60
requiredMode Group	ModeDeclarationGroup Prototype	*	aggr	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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=requiredModeGroup.shortName, required ModeGroup.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=30
requiredTrigger	Trigger	*	aggr	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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=requiredTrigger.shortName, required Trigger.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=40

Table A.121: 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	ARObject, ExecutableEntity, Identifiable, MultilanguageReferrable, Referrable			
Subclasses	BswCalledEntity, BswInterruptEntity, BswSchedulableEntity			
Aggregated by	BswInternalBehavior.entity			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
accessedMode Group	ModeDeclarationGroup Prototype	*	ref	A mode group which is accessed via API call by this entity. It shall be a ModeDeclarationGroupPrototype required by this module or cluster. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=accessedModeGroup.modeDeclaration GroupPrototype, accessedModeGroup.variation Point.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	BswModuleEntity (abstract)			
activationPoint	BswInternalTriggering Point	*	ref	Activation point used by the module entity to activate one or more internal triggers. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=activationPoint.bswInternalTriggeringPoint, activationPoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
callPoint	BswModuleCallPoint	*	aggr	A call point used in the code of this entity. 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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=callPoint.shortName, callPoint.variation Point.shortLabel vh.latestBindingTime=preCompileTime
dataReceive Point	BswVariableAccess	*	aggr	The data is received via the BSW Scheduler. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=dataReceivePoint.shortName, dataReceive Point.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
dataSendPoint	BswVariableAccess	*	aggr	The data is sent via the BSW Scheduler. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=dataSendPoint.shortName, dataSend Point.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
implemented Entry	BswModuleEntry	0..1	ref	The entry which is implemented by this module entity.
issuedTrigger	Trigger	*	ref	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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=issuedTrigger.trigger, issuedTrigger.variation Point.shortLabel vh.latestBindingTime=preCompileTime
managedMode Group	ModeDeclarationGroup Prototype	*	ref	A mode group which is managed by this entity. It shall be a ModeDeclarationGroupPrototype provided by this module or cluster. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=managedModeGroup.modeDeclaration GroupPrototype, managedModeGroup.variation Point.shortLabel vh.latestBindingTime=preCompileTime
schedulerName Prefix	BswSchedulerName Prefix	0..1	ref	A prefix to be used in generated names for the Bsw ModuleScheduler in the context of this BswModuleEntity, for example entry point prototypes, macros for dealing with exclusive areas, header file names. Details are defined in the SWS RTE. The prefix supersedes default rules for the prefix of those names.

Table A.122: BswModuleEntity

Class	BswModuleEntry
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswInterfaces
Note	This class represents a single API entry (C-function prototype) into the BSW module or cluster. 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. **Tags:**atp.recommendedPackage=BswModuleEntrys
Base	*ARElement*, ARObject, *AtpBlueprint*, AtpBlueprintable, CollectableElement, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*
Aggregated by	ARPackage.element

Attribute	Type	Mult.	Kind	Note
argument (ordered)	SwServiceArg	*	aggr	An argument belonging to this BswModuleEntry. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=argument.shortName, argument.variationPoint.shortLabel vh.latestBindingTime=blueprintDerivationTime xml.sequenceOffset=45
bswEntryKind	BswEntryKindEnum	0..1	attr	This describes whether the entry is concrete or abstract. If the attribute is missing the entry is considered as concrete. **Tags:**xml.sequenceOffset=40
callType	BswCallType	0..1	attr	The type of call associated with this service. **Tags:**xml.sequenceOffset=25
execution Context	BswExecutionContext	0..1	attr	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. **Tags:**xml.sequenceOffset=30
function Prototype Emitter	NameToken	0..1	attr	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.
isReentrant	Boolean	0..1	attr	Reentrancy from the viewpoint of function callers: • true: Enables the service to be invoked again, before the service has finished. • false: It is prohibited to invoke the service again before is has finished. **Tags:**xml.sequenceOffset=15
isSynchronous	Boolean	0..1	attr	Synchronicity from the viewpoint of function callers: • 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. **Tags:**xml.sequenceOffset=20
returnType	SwServiceArg	0..1	aggr	The return type belonging to this bswModuleEntry. **Tags:**xml.sequenceOffset=40
role	Identifier	0..1	attr	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). **Tags:**xml.sequenceOffset=10

▽

△

Class	BswModuleEntry			
serviceId	PositiveInteger	0..1	attr	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. **Tags:**xml.sequenceOffset=5
swServiceImpl Policy	SwServiceImplPolicy Enum	0..1	attr	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. **Tags:**xml.sequenceOffset=35

Table A.123: BswModuleEntry

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

Table A.124: BswModuleTiming

Class	BswOperationInvokedEvent			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	This event is thrown on operation invocation in Client-Server-Communication via the BSW Scheduler. Its "entry" reference provides the BswClientServerEntry that is called subsequently. Note this event is not needed in case of direct function calls.			
Base	*ARObject*, *AbstractEvent*, *BswEvent*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	BswInternalBehavior.event			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
entry	BswModuleClientServer Entry	0..1	ref	The providedClientServerEntry invoked by this event.

Table A.125: BswOperationInvokedEvent

Class	BswQueuedDataReceptionPolicy
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior
Note	Reception policy attributes specific for queued receiving.
Base	*ARObject*, *BswApiOptions*, *BswDataReceptionPolicy*

▽

△

Class	BswQueuedDataReceptionPolicy			
Aggregated by	BswInternalBehavior.receptionPolicy			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
queueLength	PositiveInteger	0..1	attr	Length of queue for received events.

Table A.126: BswQueuedDataReceptionPolicy

Class	BswSchedulableEntity			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	BSW module entity, which is designed for control by the BSW Scheduler. It may for example implement a so-called "main" function.			
Base	*ARObject*, *BswModuleEntity*, *ExecutableEntity*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	BswInternalBehavior.entity			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.127: BswSchedulableEntity

Class	*BswScheduleEvent* (abstract)			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	BswEvent that is able to start a BswSchedulabeEntity.			
Base	*ARObject*, *AbstractEvent*, *BswEvent*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	BswAsynchronousServerCallReturnsEvent, BswBackgroundEvent, BswDataReceivedEvent, Bsw ExternalTriggerOccurredEvent, BswInternalTriggerOccurredEvent, BswModeManagerErrorEvent, Bsw ModeSwitchEvent, BswModeSwitchedAckEvent, BswOsTaskExecutionEvent, BswTimingEvent			
Aggregated by	BswInternalBehavior.event			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.128: BswScheduleEvent

Class	BswServiceDependency			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	Specialization of ServiceDependency in the context of an BswInternalBehavior. It allows to associate BswModuleEntries and data defined for a BSW module or cluster to a given ServiceNeeds element.			
Base	*ARObject*, *ServiceDependency*			
Aggregated by	BswInternalBehavior.serviceDependency			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
assignedData	RoleBasedData Assignment	*	aggr	Defines the role of an associated data object (owned by this module or cluster) in the context of the ServiceNeeds element. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=assignedData, assignedData.variation Point.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	BswServiceDependency			
assignedEntry Role	RoleBasedBswModule EntryAssignment	*	aggr	Defines the role of an associated BswModuleEntry in the context of the ServiceNeeds element. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=assignedEntryRole, assignedEntry Role.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
ident	BswService DependencyIdent	0..1	aggr	This adds the ability to become referrable to BswService Dependency. **Stereotypes:** atpIdentityContributor **Tags:**xml.sequenceOffset=-100
serviceNeeds	ServiceNeeds	0..1	aggr	The associated ServiceNeeds.

Table A.129: BswServiceDependency

Class	BswServiceDependencyIdent			
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticMapping::ServiceMapping			
Note	This meta-class is created to add the ability to become the target of a reference to the non-Referrable BswServiceDependency.			
Base	ARObject, AtpClassifier, AtpFeature, AtpStructureElement, IdentCaption, Identifiable, Multilanguage Referrable, Referrable			
Aggregated by	AtpClassifier.atpFeature, BswServiceDependency.ident			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.130: BswServiceDependencyIdent

Class	BswSynchronousServerCallPoint			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	Represents a synchronous procedure call point via the BSW Scheduler.			
Base	ARObject, BswModuleCallPoint, Referrable			
Aggregated by	BswModuleEntity.callPoint			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
calledEntry	BswModuleClientServer Entry	0..1	ref	The entry to be called.
calledFrom WithinExclusive Area	ExclusiveAreaNesting Order	0..1	ref	This indicates that the call point is located at the deepest level inside one or more ExclusiveAreas that are nested in the given order.

Table A.131: BswSynchronousServerCallPoint

Class	BswTimingEvent			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	A recurring BswEvent driven by a time period.			
Base	*ARObject*, *AbstractEvent*, *BswEvent*, *BswScheduleEvent*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
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 A.132: BswTimingEvent

Class	BswTriggerDirectImplementation			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	Specifies a released trigger to be directly implemented via OS calls, for example in a Complex Driver module.			
Base	*ARObject*			
Aggregated by	BswInternalBehavior.triggerDirectImplementation			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
cat2Isr	Identifier	0..1	attr	The name of the OS category 2 ISR, which is controlled by the referred trigger. This means, that the module manages the category 2 ISR (e.g. according hardware initialization and enabling of ISR). Instead of calling an RTE / SchM API to raise the appropriate events in components or modules receiving the trigger, this ISR directly schedules the triggered ExecutableEntitys. The ISR name is required by the integrator to map the Bsw Events and RTEEvents to this ISR.
masteredTrigger	Trigger	0..1	ref	The trigger which is directly mastered by this module. There may be several different BswTriggerDirect Implementations mastering the same Trigger. This may be required e.g. due to memory partitioning.
task	Identifier	0..1	attr	The name of the OS task, which is controlled by the referred trigger. This means, that the module uses the trigger condition to directly activate an OS task instead of calling an API of the BswScheduler. The task name is required by the RTE generator resp. BswScheduler to raise the appropriate events in components or modules receiving the trigger.

Table A.133: BswTriggerDirectImplementation

Class	BswVariableAccess			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	The presence of a BswVariableAccess implies that a BswModuleEntity needs access to a VariableData Prototype via the BSW Scheduler. The kind of access is specified by the role in which the class is used.			
Base	*ARObject*, *Referrable*			
Aggregated by	*BswModuleEntity*.dataReceivePoint, *BswModuleEntity*.dataSendPoint			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
accessed Variable	VariableDataPrototype	0..1	ref	The data accessed via the BSW Scheduler.

$\triangledown$

△

Class	BswVariableAccess			
context Limitation	BswDistinguished Partition	*	ref	The existence of this reference indicates that the variable is received resp. sent only in the context of the referred BswDistinguishedPartitions.

Table A.134: BswVariableAccess

Class	BufferProperties			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	Configuration of the buffer properties the transformer needs to work.			
Base	ARObject			
Aggregated by	TransformationTechnology.bufferProperties			
Attribute	Type	Mult.	Kind	Note
headerLength	Integer	1	attr	Defines the length of the header (in bits) this transformer will add in front of the data.
inPlace	Boolean	1	attr	If set, the transformer uses the input buffer as output buffer.

Table A.135: BufferProperties

Class	BulkNvDataDescriptor			
Package	M2::AUTOSARTemplates::SWComponentTemplate::NvBlockComponent			
Note	This meta-class represents one bulk NV Data Block that is read-only for the application software. The purpose of a bulk NV Data Block is to provide access to information uploaded to the vehicle at e.g. the end of the production line.			
Base	ARObject, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	AtpClassifier.atpFeature, NvBlockSwComponentType.bulkNvDataDescriptor			
Attribute	Type	Mult.	Kind	Note
bulkNvBlock	VariableDataPrototype	0..1	aggr	This aggregation represents the actual bulk NVBlock.
nvBlockData Mapping	NvBlockDataMapping	*	aggr	Defines the mapping between the VariableData Prototypes in the NvBlockComponents ports and the VariableDataPrototypes of the non-volatile memory. The aggregation of NvBlockDataMapping is subject to variability with the purpose to support the conditional existence of nv data ports. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=nvBlockDataMapping, nvBlockData Mapping.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table A.136: BulkNvDataDescriptor

Class	BurstPatternEventTriggering
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint::EventTriggeringConstraint
Note	A BurstPatternEventTriggering describes the maximum number of occurrences of the same event in a given time interval. This is typically used to model a worst case activation scenario.
Base	ARObject, EventTriggeringConstraint, Identifiable, MultilanguageReferrable, Referrable, Timing Constraint, Traceable

▽

△

Class	BurstPatternEventTriggering			
Aggregated by	*TimingExtension*.timingGuarantee, *TimingExtension*.timingRequirement			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
maxNumberOf Occurrences	PositiveInteger	0..1	attr	The maximum number of event occurrences within the given time interval.
minimumInter ArrivalTime	MultidimensionalTime	0..1	aggr	Specifies the minimum distance between subsequent occurrences of the event within the given time interval.
minNumberOf Occurrences	PositiveInteger	0..1	attr	The minimum number of event occurrences within the given time interval. **Tags:**xml.sequenceOffset=10
patternJitter	MultidimensionalTime	0..1	aggr	The optional parameter "Pattern Jitter" specifies the 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 "Pattern Period".
patternLength	MultidimensionalTime	0..1	aggr	Specifies 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 optional parameter "Pattern Period" specifies the time distance between the beginnings of subsequent repetitions of the given burst pattern.

Table A.137: BurstPatternEventTriggering

Class	BusMirrorChannel			
Package	M2::AUTOSARTemplates::SystemTemplate::BusMirror			
Note	This element assigns a busMirrorNetworkId to the referenced channel.			
Base	*ARObject*			
Aggregated by	*BusMirrorChannelMapping*.sourceChannel, *BusMirrorChannelMapping*.targetChannel			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
busMirror NetworkId	PositiveInteger	1	attr	This attribute defines the networkId of the communication channel.
channel	PhysicalChannel	0..1	ref	Reference to PhysicalChannel that is used in the bus mirroring as sourceChannel or targetChannel. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=channel.physicalChannel, channel.variation Point.shortLabel vh.latestBindingTime=systemDesignTime

Table A.138: BusMirrorChannel

Class	*BusMirrorChannelMapping* (abstract)
Package	M2::AUTOSARTemplates::SystemTemplate::BusMirror
Note	This element defines a bus mirroring in which the traffic from one communication bus (sourceChannel) is forwarded to another one (targetChannel).
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*
Subclasses	BusMirrorChannelMappingCan, BusMirrorChannelMappingFlexray, BusMirrorChannelMappingIp, Bus MirrorChannelMappingUserDefined
Aggregated by	ARPackage.element

▽

△

Class	BusMirrorChannelMapping (abstract)			
Attribute	Type	Mult.	Kind	Note
sourceChannel	BusMirrorChannel	0..1	aggr	Defines the sourceChannel from which frames are received.
targetChannel	BusMirrorChannel	0..1	aggr	Defines the targetChannel to which frames are forwarded.
targetPdu Triggering	PduTriggering	*	ref	Reference to the PduTriggering that is used for transmission of the mirrored frames on the targetChannel. Please note that on FlexRay several targetPduTriggerings may be used. For all other communication channels only a single targetPduTriggering is supported. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=targetPduTriggering.pduTriggering, target PduTriggering.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.139: BusMirrorChannelMapping

Class	BusMirrorChannelMappingCan			
Package	M2::AUTOSARTemplates::SystemTemplate::BusMirror			
Note	This element defines the bus mirroring between a CAN or LIN sourceChannel and a CAN targetChannel. **Tags:**atp.recommendedPackage=BusMirrorChannelMappings			
Base	ARObject, BusMirrorChannelMapping, CollectableElement, FibexElement, Identifiable, Multilanguage Referrable, PackageableElement, Referrable			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
canIdRange Mapping	BusMirrorCanIdRange Mapping	*	aggr	Rules for remapping of a set of CAN IDs.
canIdToCanId Mapping	BusMirrorCanIdToCanId Mapping	*	aggr	Rules for remapping of single CanIds.
linPidToCanId Mapping	BusMirrorLinPidToCan IdMapping	*	aggr	Rules for remapping of single LIN Frames.
mirrorSourceLin ToCanRange BaseId	PositiveInteger	0..1	attr	Base ID merged with the LIN frame ID to form the CAN ID. Only required when a BusMirrorChannel that refers to a LinPhysicalChannel in the role channel is referenced in the role sourceChannel.
mirrorStatus CanId	PositiveInteger	0..1	attr	CAN ID of the CAN status frame. If configured, a status frame will be sent on the CAN destination bus that contains the state of all active source buses.

Table A.140: BusMirrorChannelMappingCan

Class	BusMirrorChannelMappingFlexray
Package	M2::AUTOSARTemplates::SystemTemplate::BusMirror
Note	This element defines the bus mirroring between a CAN, LIN or FlexRay sourceChannel and a FlexRay targetChannel. **Tags:**atp.recommendedPackage=BusMirrorChannelMappings
Base	ARObject, BusMirrorChannelMapping, CollectableElement, FibexElement, Identifiable, Multilanguage Referrable, PackageableElement, Referrable

▽

△

Class	BusMirrorChannelMappingFlexray			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
transmission Deadline	TimeValue	0..1	attr	Time in seconds after which the collection of source frames into the destination frame is stopped and the frame is sent at the latest. If omitted, destination frames are only sent when full or when the time stamp overflows.

Table A.141: BusMirrorChannelMappingFlexray

Class	BusMirrorChannelMappingIp			
Package	M2::AUTOSARTemplates::SystemTemplate::BusMirror			
Note	This element defines the bus mirroring between a CAN, LIN or FlexRay sourceChannel and an Ethernet IP targetChannel. **Tags:**atp.recommendedPackage=BusMirrorChannelMappings			
Base	*ARObject*, *BusMirrorChannelMapping*, *CollectableElement*, *FibexElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
transmission Deadline	TimeValue	0..1	attr	Time in seconds after which the collection of source frames into the destination frame is stopped and the frame is sent at the latest. If omitted, destination frames are only sent when full or when the time stamp overflows.

Table A.142: BusMirrorChannelMappingIp

Class	BusMirrorChannelMappingUserDefined			
Package	M2::AUTOSARTemplates::SystemTemplate::BusMirror			
Note	This element defines the bus mirroring between a CAN, LIN or FlexRay sourceChannel and a User Defined targetChannel. **Tags:**atp.recommendedPackage=BusMirrorChannelMappings			
Base	*ARObject*, *BusMirrorChannelMapping*, *CollectableElement*, *FibexElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
transmission Deadline	TimeValue	0..1	attr	Time in seconds after which the collection of source frames into the destination frame is stopped and the frame is sent at the latest. If omitted, destination frames are only sent when full or when the time stamp overflows.

Table A.143: BusMirrorChannelMappingUserDefined

Class	*BusspecificNmEcu* (abstract)
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement
Note	Busspecific NmEcu attributes.
Base	*ARObject*

▽

△

Class	BusspecificNmEcu (abstract)			
Subclasses	CanNmEcu, FlexrayNmEcu, J1939NmEcu, UdpNmEcu			
Aggregated by	NmEcu.busDependentNmEcu			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.144: BusspecificNmEcu

Enumeration	ByteOrderEnum
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::PrimitiveTypes
Note	When more than one byte is stored in the memory the order of those bytes may differ depending on the architecture of the processing unit. If the least significant byte is stored at the lowest address, this architecture is called little endian and otherwise it is called big endian. ByteOrder is very important in case of communication between different PUs or ECUs.
Aggregated by	ApSomeipTransformationProps.byteOrder, BaseTypeDirectDefinition.byteOrder, DiagnosticCommon Props.defaultEndianness, ISignalToIPduMapping.packingByteOrder, MultiplexedIPdu.selectorField ByteOrder, PduToFrameMapping.packingByteOrder, SegmentPosition.segmentByteOrder, SOMEIP TransformationDescription.byteOrder, System.containerIPduHeaderByteOrder
Literal	Description
mostSignificantByte First	Most significant byte shall come at the lowest address (also known as BigEndian or as Motorola-Format) **Tags:**atp.EnumerationLiteralIndex=0
mostSignificantByte Last	Most significant byte shall come highest address (also known as LittleEndian or as Intel-Format) **Tags:**atp.EnumerationLiteralIndex=1
opaque	For opaque data endianness conversion has to be configured to Opaque. See AUTOSAR COM Specification for more details. **Tags:**atp.EnumerationLiteralIndex=2

Table A.145: ByteOrderEnum

Class	CalibrationParameterValue			
Package	M2::AUTOSARTemplates::SWComponentTemplate::MeasurementAndCalibration::CalibrationParameter Values			
Note	Specifies instance specific calibration parameter values used to initialize the memory objects implementing calibration parameters in the generated RTE code. RTE generator will use the implInitValue to override the initial values specified for the DataPrototypes of a component type. The applInitValue is used to exchange init values with the component vendor not publishing the transformation algorithm between ApplicationDataTypes and ImplementationDataTypes or defining an instance specific initialization of components which are only defined with ApplicationDataTypes. Note: If both representations of init values are available these need to represent the same content. Note further that in this case an explicit mapping of ValueSpecification is not implemented because calibration parameters are delivered back after the calibration phase.			
Base	ARObject			
Aggregated by	CalibrationParameterValueSet.calibrationParameterValue			
Attribute	Type	Mult.	Kind	Note
applInitValue	ValueSpecification	0..1	aggr	This is the initial value specification structured according to the ApplicationDataType

▽

$\triangle$

Class	CalibrationParameterValue			
implInitValue	ValueSpecification	0..1	aggr	This is the initial value specification structured according to the ImplementationDataType
initialized Parameter	FlatInstanceDescriptor	0..1	ref	This represents the parameter that is initialized by the CalibrationParameterValue.

Table A.146: CalibrationParameterValue

Enumeration	CalprmAxisCategoryEnum
Package	M2::MSR::DataDictionary::CalibrationParameter
Note	This enum specifies the possible values of the category property within SwCalprmAxis.
Aggregated by	RuleBasedAxisCont.category, SwAxisCont.category, SwCalprmAxis.category
Literal	**Description**
comAxis	COM_AXIS is equal to an STD_AXIS, the difference is, that a COM_AXIS is an shared axis, that means this axis can be used multiple times by different CURVEs, MAPs, CUBOIDs, CUBE_4s, and CUBE_5s. **Tags:** atp.EnumerationLiteralIndex=0 xml.name=COM_AXIS
fixAXIS	FIX_AXIS means that the input axis is not stored. The axis is calculated using parameters and so on it is also not possible to modify the axis points. **Tags:** atp.EnumerationLiteralIndex=4 xml.name=FIX_AXIS
resAxis	RES_AXIS is also an shared axis like COM_AXIS, the difference is that this kind of axis can be used for rescaling. **Tags:** atp.EnumerationLiteralIndex=6 xml.name=RES_AXIS
stdAxis	STD_AXIS means that input and output axis definition are stored within this CURVE, MAP, CUBOID, CUBE_4, and CUBE_5. There is no shared or calculated axis. **Tags:** atp.EnumerationLiteralIndex=8 xml.name=STD_AXIS

Table A.147: CalprmAxisCategoryEnum

Class	CanCommunicationConnector			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	CAN bus specific communication connector attributes.			
Base	ARObject, AbstractCanCommunicationConnector, CommunicationConnector, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	EcuInstance.connector, MachineDesign.communicationConnector			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
pncWakeupCanId	PositiveInteger	0..1	attr	CAN Identifier used to configure the CAN Transceiver for partial network wakeup.
pncWakeupCanIdExtended	Boolean	0..1	attr	Defines whether pncWakeupCanId and pncWakeupCanId Mask shall be interpreted as extended or standard CAN ID.

$\triangledown$

△

Class	CanCommunicationConnector			
pncWakeupCanIdMask	PositiveInteger	0..1	attr	Bit mask for CAN Identifier used to configure the CAN Transceiver for partial network wakeup.
pncWakeupDataMask	PositiveUnlimitedInteger	0..1	attr	Bit mask for CAN Payload used to configure the CAN Transceiver for partial network wakeup.
pncWakeupDlc	PositiveInteger	0..1	attr	Data Length of the remote data frame used to configure the CAN Transceiver for partial network wakeup in Bytes.

Table A.148: CanCommunicationConnector

Class	<<atpVariation>> **CanCommunicationController**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	CAN bus specific communication port attributes.			
Base	*ARObject*, *AbstractCanCommunicationController*, *CommunicationController*, *Identifiable*, *Multilanguage Referrable*, *Referrable*			
Aggregated by	EcuInstance.commController, MachineDesign.communicationController			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.149: CanCommunicationController

Class	CanControllerConfiguration			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	This element is used for the specification of the exact CAN Bit Timing configuration parameter values.			
Base	*ARObject*, *AbstractCanCommunicationControllerAttributes*			
Aggregated by	*AbstractCanCommunicationController*.canControllerAttributes, CanXlProps.canConfig			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
propSeg	Integer	0..1	attr	Specifies propagation delay in time quantas.
syncJumpWidth	Integer	1	attr	The number of quanta in the Synchronization Jump Width, SJW. The (Re-)Synchronization Jump Width (SJW) defines how far a resynchronization may move the Sample Point inside the limits defined by the Phase Buffer Segments to compensate for edge phase errors.
timeSeg1	Integer	1	attr	Specifies phase segment 1 in time quantas. timeSeg1 = Phase_Seg1
timeSeg2	Integer	1	attr	Specifies phase segment 2 in time quantas. timeSeg2 = Phase_Seg2

Table A.150: CanControllerConfiguration

Class	CanControllerConfigurationRequirements			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	This element allows the specification of ranges for the CAN Bit Timing configuration parameters. These ranges are taken as requirements and have to be respected by the ECU developer.			
Base	*ARObject*, *AbstractCanCommunicationControllerAttributes*			
Aggregated by	*AbstractCanCommunicationController*.canControllerAttributes			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	CanControllerConfigurationRequirements			
maxNumberOfTimeQuantaPerBit	Integer	0..1	attr	Maximum number of time quanta in the bit time.
maxSamplePoint	Float	0..1	attr	The max. value of the sample point as a percentage of the total bit time.
maxSyncJumpWidth	Float	0..1	attr	The max. Synchronization Jump Width value as a percentage of the total bit time. The (Re-)Synchronization Jump Width (SJW) defines how far a resynchronization may move the Sample Point inside the limits defined by the Phase Buffer Segments to compensate for edge phase errors.
minNumberOfTimeQuantaPerBit	Integer	0..1	attr	Minimum number of time quanta in the bit time.
minSamplePoint	Float	0..1	attr	The min. value of the sample point as a percentage of the total bit time.
minSyncJumpWidth	Float	0..1	attr	The min. Synchronization Jump Width value as a percentage of the total bit time. The (Re-)Synchronization Jump Width (SJW) defines how far a resynchronization may move the Sample Point inside the limits defined by the Phase Buffer Segments to compensate for edge phase errors.

Table A.151: CanControllerConfigurationRequirements

Class	CanControllerFdConfiguration			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	Bit timing related configuration of a CAN controller for payload and CRC of a CAN FD frame.			
Base	ARObject			
Aggregated by	*AbstractCanCommunicationControllerAttributes*.canControllerFdAttributes, CanXlProps.canFdConfig			
Attribute	Type	Mult.	Kind	Note
paddingValue	PositiveInteger	0..1	attr	Specifies the value which is used to pad unused data in CAN FD frames which are bigger than 8 byte if the length of a Pdu which was requested to be sent does not match the allowed DLC values of CAN FD.
propSeg	PositiveInteger	1	attr	Specifies propagation delay in time quantas.
sspOffset	PositiveInteger	0..1	attr	Specifies the Transmitter Delay Compensation Offset in minimum time quanta. Transmitter Delay Compensation Offset is used to adjust the position of the Secondary Sample Point (SSP), relative to the beginning of the received bit. If this parameter is configured, the Transmitter Delay Compensation is done by measurement of the CAN controller. If not specified Transmitter Delay Compensation is disabled.
syncJumpWidth	PositiveInteger	1	attr	Specifies the synchronization jump width for the controller in time quantas.
timeSeg1	PositiveInteger	1	attr	Specifies phase segment 1 in time quantas.
timeSeg2	PositiveInteger	1	attr	Specifies phase segment 2 in time quantas.

▽

△

Class	CanControllerFdConfiguration			
txBitRateSwitch	Boolean	1	attr	Specifies if the bit rate switching shall be used for transmissions. TRUE: CAN FD frames shall be sent with bit rate switching. FALSE: CAN FD frames shall be sent without bit rate switching.

Table A.152: CanControllerFdConfiguration

Class	CanControllerFdConfigurationRequirements			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	This element allows the specification of ranges for the CanFD bit timing configuration parameters. These ranges are taken as requirements and shall be respected by the ECU developer.			
Base	*ARObject*			
Aggregated by	*AbstractCanCommunicationControllerAttributes*.canControllerFdRequirements			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
maxNumberOf TimeQuantaPer Bit	Integer	0..1	attr	Maximum number of time quanta in the bit time.
maxSample Point	Float	0..1	attr	The max. value of the sample point as a percentage of the total bit time.
maxSyncJump Width	Float	0..1	attr	The max. Synchronization Jump Width value as a percentage of the total bit time. The (Re-)Synchronization Jump Width (SJW) defines how far a resynchronization may move the Sample Point inside the limits defined by the Phase Buffer Segments to compensate for edge phase errors.
maxTrcvDelay Compensation Offset	TimeValue	0..1	attr	Specifies the maximum Transceiver Delay Compensation Offset in seconds. If not specified Transceiver Delay Compensation is disabled.
minNumberOf TimeQuantaPer Bit	Integer	0..1	attr	Minimum number of time quanta in the bit time.
minSamplePoint	Float	0..1	attr	The min. value of the sample point as a percentage of the total bit time.
minSyncJump Width	Float	0..1	attr	The min. Synchronization Jump Width value as a percentage of the total bit time. The (Re-)Synchronization Jump Width (SJW) defines how far a resynchronization may move the Sample Point inside the limits defined by the Phase Buffer Segments to compensate for edge phase errors.
minTrcvDelay Compensation Offset	TimeValue	0..1	attr	Specifies the minimum Transceiver Delay Compensation Offset in seconds. If not specified Transceiver Delay Compensation is disabled.
paddingValue	PositiveInteger	0..1	attr	Specifies the value which is used to pad unused data in CAN FD frames which are bigger than 8 byte if the length of a Pdu which was requested to be sent does not match the allowed DLC values of CAN FD.

▽

△

Class	CanControllerFdConfigurationRequirements			
txBitRateSwitch	Boolean	0..1	attr	Specifies if the bit rate switching shall be used for transmissions. TRUE: CAN FD frames shall be sent with bit rate switching. FALSE: CAN FD frames shall be sent without bit rate switching.

Table A.153: CanControllerFdConfigurationRequirements

Class	CanControllerXlConfiguration			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	This meta-class represents the CAN XL-specific controller attributes.			
Base	*ARObject*			
Aggregated by	*AbstractCanCommunicationControllerAttributes*.canControllerXlAttributes, CanXlProps.canXlConfig			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
errorSignaling Enabled	Boolean	0..1	attr	Specifies if error signaling shall be enabled. This is not possible when the transceiver is switched to PWM mode (trcvPwmModeEnabled set to TRUE). TRUE: Error signaling shall be enabled. FALSE: Error signaling shall be disabled.
propSeg	PositiveInteger	0..1	attr	Specifies propagation delay in time quantas.
pwmL	PositiveInteger	0..1	attr	Specifies the PWM long phase length.
pwmO	PositiveInteger	0..1	attr	Specifies the PWM time offset.
pwmS	PositiveInteger	0..1	attr	Specifies the PWM short phase length.
sspOffset	PositiveInteger	0..1	attr	Specifies the Transmitter Delay Compensation Offset in minimum time quanta. Transmitter Delay Compensation Offset is used to adjust the position of the Secondary Sample Point (SSP), relative to the beginning of the received bit. If this parameter is configured, the Transmitter Delay Compensation is done by measurement of the CAN controller. If not specified Transmitter Delay Compensation is disabled.
syncJumpWidth	PositiveInteger	0..1	attr	Specifies the synchronization jump width for the controller in time quantas.
timeSeg1	PositiveInteger	0..1	attr	Specifies phase segment 1 in time quantas.
timeSeg2	PositiveInteger	0..1	attr	Specifies phase segment 2 in time quantas.
trcvPwmMode Enabled	Boolean	0..1	attr	Specifies if the transceiver shall be set to the PWM mode. TRUE: The transceiver shall be switched to PWM mode. FALSE: The transceiver shall work in classic CAN mode.

Table A.154: CanControllerXlConfiguration

Class	CanControllerXlConfigurationRequirements
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology
Note	This element allows the specification of ranges for the CAN XL configuration parameters. These ranges are taken as requirements and have to be respected by the ECU developer.
Base	*ARObject*

▽

△

Class	CanControllerXlConfigurationRequirements			
Aggregated by	*AbstractCanCommunicationControllerAttributes*.canControllerXlRequirements, CanXlProps.canXlConfigReqs			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
errorSignaling Enabled	Boolean	0..1	attr	Specifies if error signaling shall be enabled. This is not possible when the transceiver is switched to PWM mode (trcvPwmModeEnabled set to TRUE). TRUE: Error signaling shall be enabled. FALSE: Error signaling shall be disabled.
maxNumberOf TimeQuantaPer Bit	Integer	0..1	attr	Maximum number of time quanta in the bit time.
maxPwmL	PositiveInteger	0..1	attr	Specifies the maximum PWM long phase length.
maxPwmO	PositiveInteger	0..1	attr	Specifies the minimum PWM time offset.
maxPwmS	PositiveInteger	0..1	attr	Specifies the maximum PWM short phase length.
maxSample Point	Float	0..1	attr	The max. value of the sample point as a percentage of the total bit time.
maxSyncJump Width	Float	0..1	attr	The max. Synchronization Jump Width value as a percentage of the total bit time. The (Re-)Synchronization Jump Width (SJW) defines how far a resynchronization may move the Sample Point inside the limits defined by the Phase Buffer Segments to compensate for edge phase errors.
maxTrcvDelay Compensation Offset	TimeValue	0..1	attr	Specifies the maximum Transceiver Delay Compensation Offset in seconds. If not specified Transceiver Delay Compensation is disabled.
minNumberOf TimeQuantaPer Bit	Integer	0..1	attr	Minimum number of time quanta in the bit time.
minPwmL	PositiveInteger	0..1	attr	Specifies the minimum PWM long phase length.
minPwmO	PositiveInteger	0..1	attr	Specifies the maximum PWM time offset.
minPwmS	PositiveInteger	0..1	attr	Specifies the minimum PWM short phase length.
minSamplePoint	Float	0..1	attr	The min. value of the sample point as a percentage of the total bit time.
minSyncJump Width	Float	0..1	attr	The min. Synchronization Jump Width value as a percentage of the total bit time. The (Re-)Synchronization Jump Width (SJW) defines how far a resynchronization may move the Sample Point inside the limits defined by the Phase Buffer Segments to compensate for edge phase errors.
minTrcvDelay Compensation Offset	TimeValue	0..1	attr	Specifies the minimum Transceiver Delay Compensation Offset in seconds. If not specified Transceiver Delay Compensation is disabled.
trcvPwmMode Enabled	Boolean	0..1	attr	Specifies if the transceiver shall be set to the PWM mode. TRUE: The transceiver shall be switched to PWM mode. FALSE: The transceiver shall work in classic CAN mode.

Table A.155: CanControllerXlConfigurationRequirements

Document ID 635: AUTOSAR_TR_ClassicAutosarModelConstraints

Class	CanFrameTriggering			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanCommunication			
Note	CAN specific attributes to the FrameTriggering			
Base	ARObject, FrameTriggering, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	PhysicalChannel.frameTriggering			
Attribute	Type	Mult.	Kind	Note
absolutely Scheduled Timing	TtcanAbsolutely ScheduledTiming	*	aggr	Each frame in TTCAN is identified by its slot id and communication cycle. A description is provided by the usage of AbsolutelyScheduledTiming.
canAddressing Mode	CanAddressingMode Type	1	attr	The CAN protocol supports two types of frame formats. The standard frame format uses 11-bit identifiers and is defined in the CAN specification 2.0 A. Additionally the extended frame format allows 29-bit identifiers and is defined in the CAN specification 2.0 B.
canFrameRx Behavior	CanFrameRxBehavior Enum	0..1	attr	Defines which CAN protocol shall be expected for frame reception.
canFrameTx Behavior	CanFrameTxBehavior Enum	0..1	attr	Defines which CAN protocol shall be used for frame transmission.
canXlFrame TriggeringProps	CanXlFrameTriggering Props	0..1	aggr	Definition of CAN XL specific attributes in case the frame is a CAN XL frame.
identifier	Integer	0..1	attr	This attribute is used to define the identifier this frame shall use on the CAN network.
j1939requestable	Boolean	0..1	attr	Frame can be triggered by the J1939 request message.
rxIdentifier Range	RxIdentifierRange	0..1	aggr	Optional definition of a CanId range.
rxMask	PositiveInteger	0..1	attr	Identifier mask which denotes the relevant bits in the CAN Identifier. Together with the identifier, this parameter defines a CAN identifier range.
txMask	PositiveInteger	0..1	attr	Identifier mask which denotes static bits in the CAN identifier. The other bits can be set dynamically.

Table A.156: CanFrameTriggering

Enumeration	CanFrameTxBehaviorEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanCommunication
Note	Defines different CAN protocols for frame transmission behavior.
Aggregated by	CanFrameTriggering.canFrameTxBehavior
Literal	Description
can20	This CAN frame shall be sent as CAN 2.0 only. **Tags:**atp.EnumerationLiteralIndex=0
canFd	This CAN frame shall be sent as CAN FD. **Tags:**atp.EnumerationLiteralIndex=1

Table A.157: CanFrameTxBehaviorEnum

Class	CanNmCluster			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	Can specific NmCluster attributes			
Base	ARObject, Identifiable, MultilanguageReferrable, NmCluster, Referrable			
Aggregated by	NmConfig.nmCluster			
Attribute	Type	Mult.	Kind	Note

▽

△

Class	CanNmCluster				
nmBusload ReductionActive	Boolean	1	attr	It determines if bus load reduction for the respective Can Nm channel is active or not.
nmCarWakeUp BitPosition	PositiveInteger	0..1	attr	Specifies the bit position of the CarWakeUp within the Nm Pdu.
nmCarWakeUp FilterNodeId	PositiveInteger	0..1	attr	Source node identifier for CarWakeUp filtering.
nmCbvPosition	Integer	0..1	attr	Defines the position of the control bit vector within the Nm Pdu (Byte position). If this attribute is not configured, the Control Bit Vector is not used.
nmImmediate NmCycleTime	TimeValue	0..1	attr	Defines the immediate NmPdu cycle time in seconds which is used for nmImmediateNmTransmissions NmPdu transmissions. This parameter is only valid if CanNm ImmediateNmTransmissions is greater one.
nmImmediate Nm Transmissions	PositiveInteger	1	attr	Defines the number of immediate NmPdus which shall be transmitted. If the value is zero no immediate NmPdus are transmitted. The cycle time of immediate NmPdus is defined by nmImmediateNmCycleTime.
nmMessage TimeoutTime	TimeValue	1	attr	Timeout of an NmPdu in seconds. It determines how long the NM shall wait with notification of transmission failure while communication errors occur on the bus.
nmMsgCycle Time	TimeValue	1	attr	Period of a NmPdu in seconds. It determines the periodic rate in the periodic transmission mode with bus load reduction and is the basis for transmit scheduling in the periodic transmission mode without bus load reduction.
nmNetwork Timeout	TimeValue	1	attr	Network Timeout for NmPdus in seconds It denotes the time how long the CanNm shall stay in the Network Mode before transition into Prepare Bus-Sleep Mode shall take place.
nmNidPosition	Integer	0..1	attr	Defines the byte position of the source node identifier within the NmPdu. If this attribute is not configured, the Node Identification is not used.
nmRemote SleepIndication Time	TimeValue	1	attr	Timeout for Remote Sleep Indication in seconds. It defines the time how long it shall take to recognize that all other nodes are ready to sleep.
nmRepeat MessageTime	TimeValue	1	attr	Timeout for Repeat Message State in seconds. Defines the time how long the NM shall stay in the Repeat Message State.
nmWaitBus SleepTime	TimeValue	1	attr	Timeout for bus calm down phase in seconds. It denotes the time how long the CanNm shall stay in the Prepare Bus-Sleep Mode before transition into Bus-Sleep Mode shall take place.

Table A.158: CanNmCluster

Class	CanPhysicalChannel			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	CAN bus specific physical channel attributes.			
Base	*ARObject*, *AbstractCanPhysicalChannel*, *Identifiable*, *MultilanguageReferrable*, *PhysicalChannel*, *Referrable*			
Aggregated by	*CommunicationCluster*.physicalChannel			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.159: CanPhysicalChannel

Class	CanTpConnection			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	A connection identifies the sender and the receiver of this particular communication. The CanTp module routes a Pdu through this connection.			
	atpVariation: Derived, because TpNode can vary.			
Base	*ARObject*, *TpConnection*			
Aggregated by	CanTpConfig.tpConnection			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
addressing Format	CanTpAddressing FormatType	1	attr	Declares which communication addressing mode is supported.
cancellation	Boolean	0..1	attr	With this switch Tx Cancellation can be turned on or off. Please note that the Rx Cancellation is always enabled.
canTpChannel	CanTpChannel	1	ref	Reference to the CanTpChannel on which this CanTp Connection is realized.
dataPdu	NPdu	1	ref	Reference to an Data NPdu.
flowControlPdu	NPdu	0..1	ref	Reference to the Flow Control NPdu.
maxBlockSize	Integer	0..1	attr	The maximum number of N-PDUs the CanTp receiver allows the sender to send, before waiting for an authorization to continue transmission of the following N-PDUs. For further details on this parameter value see ISO 15765-2 specification.
				Note: For reasons of buffer length, the CAN Transport Layer can adapt the BS value within the limit of this maximum BS
multicast	CanTpAddress	0..1	ref	TP address for 1:n connections.
padding Activation	Boolean	1	attr	This specifies whether or not Sfs, FCs and the last CF shall be padded to 8 bytes length in case it contains less payload.
				true: The N-PDU received uses padding for SF, FC and the last CF. (N-PDU length is always 8 bytes)
				false: The N-PDU received does not use padding for SF, CF and the last CF. (N-PDU length is dynamic)
receiver	CanTpNode	*	ref	The target of the TP connection.
taType	NetworkTargetAddress Type	0..1	attr	Network Target Address type.
timeoutBr	TimeValue	0..1	attr	Value in seconds of the performance requirement for (N_ Br + N_Ar). N_Br is the elapsed time between the receiving indication of a FF or CF or the transmit confirmation of a FC, until the transmit request of the next FC.
timeoutBs	TimeValue	0..1	attr	This parameter defines the timeout for waiting for an FC or AF on the sender side in an 1:1 connection. Specified in seconds.
timeoutCr	TimeValue	0..1	attr	This parameter defines the timeout value for waiting for a CF or FF-x (in case of retry) after receiving the last CF or after sending an FC or AF on the receiver side. Specified in seconds.
timeoutCs	TimeValue	0..1	attr	The attribute timeoutCs represents the time (in seconds) which elapses between the transmit request of a CF N-PDU until the transmit request of the next CF N-PDU.
tpSdu	IPdu	1	ref	Reference to an IPdu that is segmented by the Transport Protocol.
transmitter	CanTpNode	0..1	ref	The source of the TP connection.

Table A.160: CanTpConnection

Class	CanXlFrameTriggeringProps			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanCommunication			
Note	This element indicates the frame being CAN XL and contains further CAN XL specific attributes.			
Base	*ARObject*			
Aggregated by	CanFrameTriggering.canXlFrameTriggeringProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
acceptance Field	PositiveInteger	0..1	attr	Acceptance field of a CAN XL message.
priorityId	PositiveInteger	0..1	attr	Priority ID of a CAN XL message.
sduType	PositiveInteger	0..1	attr	SDU type of a CAN XL message.
vcid	PositiveInteger	0..1	attr	Virtual CAN network ID of a CAN XL message.

Table A.161: CanXlFrameTriggeringProps

Class	Caption			
Package	M2::MSR::Documentation::BlockElements			
Note	This meta-class represents the ability to express a caption which is a title, and a shortName.			
Base	*ARObject*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	MlFigure.figureCaption, MlFormula.formulaCaption, Table.tableCaption			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
desc	MultiLanguageOverview Paragraph	0..1	aggr	This represents a general but brief (one paragraph) description what the object in question is about. It is only one paragraph! This property helps a human reader to identify the object in question. **Tags:**xml.sequenceOffset=10

Table A.162: Caption

Class	Chapter			
Package	M2::MSR::Documentation::Chapters			
Note	This meta-class represents a chapter of a document. Chapters are the primary structuring element in documentation.			
Base	*ARObject*, *DocumentViewSelectable*, *Identifiable*, *MultilanguageReferrable*, *Paginateable*, *Referrable*			
Aggregated by	ChapterOrMsrQuery.chapter, MsrQueryResultChapter.chapter, SwComponentDocumentation.chapter, SwComponentDocumentation.swCalibrationNotes, SwComponentDocumentation.swCarbDoc, Sw ComponentDocumentation.swDiagnosticsNotes, SwComponentDocumentation.swFeatureDef, Sw ComponentDocumentation.swFeatureDesc, SwComponentDocumentation.swMaintenanceNotes, Sw ComponentDocumentation.swTestDesc, System.systemDocumentation			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
chapterModel	ChapterModel	1	aggr	This represents the overall contents of the chapter. **Tags:** xml.roleElement=false xml.roleWrapperElement=false xml.typeElement=false xml.typeWrapperElement=false

$\triangledown$

Document ID 635: AUTOSAR_TR_ClassicAutosarModelConstraints

△

Class	Chapter			
helpEntry	String	0..1	attr	This specifies an entry point in an online help system to be linked with the parent class. The syntax shall be defined by the applied help system respectively help system generator. Maybe it is a concatenated Identifier, but as of now we leave it as an arbitrary string. **Tags:**xml.attribute=true

Table A.163: Chapter

Class	ClassContentConditional			
Package	M2::AUTOSARTemplates::CommonStructure::StandardizationTemplate::DataExchangePoint::Data FormatTailoring			
Note	Specifies the valid content of the class. The content can optionally depend on a condition. (E.g. value of attribute 'category')			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*ClassTailoring*.classContent			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
attribute Tailoring	AttributeTailoring	*	aggr	Tailorings of the owned and inherited attributes of this Meta Classes **Tags:**xml.sequenceOffset=20
condition	AbstractCondition	0..1	aggr	The rules on the content of this class are enabled if the condition validates to true. **Tags:**xml.sequenceOffset=10
constraint Tailoring	ConstraintTailoring	*	aggr	Specification of tailorings of Constraints of that are owned by this Meta Classes **Tags:**xml.sequenceOffset=30
sdgTailoring	SdgTailoring	*	aggr	Specification of the applicable Special Data Group **Tags:**xml.sequenceOffset=40

Table A.164: ClassContentConditional

Class	*ClassTailoring* (abstract)			
Package	M2::AUTOSARTemplates::CommonStructure::StandardizationTemplate::DataExchangePoint::Data FormatTailoring			
Note	The ClassTailoring is an abstract class that allows the tailoring of its attributes, applicable constraints and Sdgs.			
Base	*ARObject*			
Subclasses	AbstractClassTailoring, ConcreteClassTailoring			
Aggregated by	AggregationTailoring.typeTailoring, DataFormatTailoring.classTailoring, ReferenceTailoring.typeTailoring			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
classContent	ClassContent Conditional	*	aggr	Specifies the accepted / not accepted content of the class. All rules apply that fullfill the condition of the Class ContentConditional **Tags:**xml.sequenceOffset=30

▽

△

Class	ClassTailoring (abstract)			
multiplicity Restriction	MultiplicityRestriction WithSeverity	0..1	aggr	Specifies the multiplicity of the class in the current context. **Tags:**xml.sequenceOffset=10
variation Restriction	VariationRestrictionWith Severity	0..1	aggr	Specifies restrictions on the usage of variant handling. **Tags:**xml.sequenceOffset=20

Table A.165: ClassTailoring

Class	ClientComSpec			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	Client-specific communication attributes (RPortPrototype typed by ClientServerInterface).			
Base	ARObject, RPortComSpec			
Aggregated by	AbstractRequiredPortPrototype.requiredComSpec, PortPrototypeBlueprint.requiredComSpec			
Attribute	Type	Mult.	Kind	Note
endToEndCall Response Timeout	TimeValue	0..1	attr	This attribute defines the maximum time interval in which the application shall expect the servers's response (time between the sending of the call invocation until the arrival of the server's response).
operation	ClientServerOperation	0..1	ref	This represents the corresponding ClientServerOperation.
transformation ComSpecProps	TransformationCom SpecProps	*	aggr	This references the TransformationComSpecProps which define port-specific configuration for data transformation.

Table A.166: ClientComSpec

Class	ClientIdDefinition			
Package	M2::AUTOSARTemplates::SystemTemplate			
Note	Several clients in one client-ECU can communicate via inter-ECU client-server communication with a server on a different ECU, if a client identifier is used to distinguish the different clients. The Client Identifier of the transaction handle that is used by the RTE can be defined by this element.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	ClientIdDefinitionSet.clientIdDefinition			
Attribute	Type	Mult.	Kind	Note
clientId	Numerical	1	attr	The Client Identifier of the transaction handle used for an inter-ECU client server communication is defined by this attribute. If defined the RTE generator shall use this client Id.
clientServer Operation	ClientServerOperation	1	iref	Reference to the ClientServerOperation that is called by the client. **InstanceRef implemented by:**OperationInSystem InstanceRef

Table A.167: ClientIdDefinition

Class	ClientIdRange
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology
Note	With this element it is possible to restrict the Client Identifier of the transaction handle that is generated by the client RTE for inter-Ecu Client/Server communication to an allowed range of numerical values.
Base	ARObject

▽

Document ID 635: AUTOSAR_TR_ClassicAutosarModelConstraints

△

Class	ClientServerInterface			
operation	ClientServerOperation	*	aggr	ClientServerOperation(s) of this ClientServerInterface. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=operation.shortName, operation.variationPoint.shortLabel vh.latestBindingTime=blueprintDerivationTime
possibleError	ApplicationError	*	aggr	Application errors that are defined as part of this interface.

Table A.171: ClientServerInterface

Class	ClientServerInterfaceMapping			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	Defines the mapping of ClientServerOperations in context of two different ClientServerInterfaces.			
Base	ARObject, *AtpBlueprint*, AtpBlueprintable, *Identifiable*, *MultilanguageReferrable*, *PortInterfaceMapping*, *Referrable*			
Aggregated by	PortInterfaceMappingSet.portInterfaceMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
errorMapping	ClientServerApplication ErrorMapping	*	aggr	Map two different ApplicationErrors defined in the context of two different ClientServerInterfaces.
operation Mapping	ClientServerOperation Mapping	*	aggr	Mapping of two ClientServerOperations in two different ClientServerInterfaces

Table A.172: ClientServerInterfaceMapping

Class	ClientServerInterfaceToBswModuleEntryBlueprintMapping			
Package	M2::AUTOSARTemplates::CommonStructure::StandardizationTemplate::ClientServerInterfaceToBsw ModuleEntryMapping			
Note	This represents a mapping between one ClientServerInterface blueprint and BswModuleEntry blueprint in order to express the intended implementation of ClientServerOperations by specific BswModuleEntries under consideration of PortDefinedArguments. Such a mapping enables the formal check whether the number of arguments and the data types of arguments of the operation + additional PortDefined Arguments matches the signature of the BswModuleEntry. **Tags:**atp.recommendedPackage=BlueprintMappingSets			
Base	*ARElement*, ARObject, *AtpBlueprint*, CollectableElement, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
clientServer Interface	ClientServerInterface	1	ref	The referenced ClientServerInterface represents the client server interface the mapping is dedicated to.
operation Mapping	ClientServerOperation BlueprintMapping	1..*	aggr	This specifies the operations used in the mapping between the ClientServerInterface and the BswModule Entry. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=operationMapping, operation Mapping.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	ClientServerInterfaceToBswModuleEntryBlueprintMapping				
portDefined Argument Blueprint (ordered)	PortDefinedArgument Blueprint	*	aggr	This specifies the PortDefinedArguments used in the mapping between the ClientServerInterface and the Bsw ModuleEntry. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=portDefinedArgumentBlueprint, portDefined ArgumentBlueprint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime	

Table A.173: ClientServerInterfaceToBswModuleEntryBlueprintMapping

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, Diagnostic DataElementInterface.read, DiagnosticDataIdentifierInterface.read, DiagnosticDataIdentifierInterface. write, DiagnosticRoutineInterface.requestResult, DiagnosticRoutineInterface.start, DiagnosticRoutine Interface.stop, PhmRecoveryActionInterface.recovery, ServiceInterface.method			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
argument (ordered)	ArgumentDataPrototype	*	aggr	An argument of this ClientServerOperation **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=argument.shortName, argument.variation Point.shortLabel vh.latestBindingTime=blueprintDerivationTime
diagArgIntegrity	Boolean	0..1	attr	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. 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. If the attribute does not exist or is set to false the Client ServerOperation does not have to consider the usage of a shared buffer.
possibleError	ApplicationError	*	ref	Possible errors that may by raised by the referring operation.

Table A.174: ClientServerOperation

Class	ClientServerOperationBlueprintMapping
Package	M2::AUTOSARTemplates::CommonStructure::StandardizationTemplate::ClientServerInterfaceToBsw ModuleEntryMapping
Note	This class describes a specific mapping between a ClientServerOperation in a ClientServerInterface blueprint and a BswModuleEntry blueprint.
Base	*ARObject*
Aggregated by	ClientServerInterfaceToBswModuleEntryBlueprintMapping.operationMapping

▽

△

Class	ClientServerOperationBlueprintMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
blueprint MappingGuide	DocumentationBlock	0..1	aggr	This attribute offers the possibility to provide additional information with respect to the mapping.
bswModule Entry	BswModuleEntry	1	ref	The referenced BswModuleEntry represents the Bsw ModuleEntry the mapping is dedicated to.
clientServer Operation	ClientServerOperation	1	ref	The referenced ClientServerOperation represents the client server operation the mapping is dedicated to.

Table A.175: ClientServerOperationBlueprintMapping

Class	ClientServerOperationMapping			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	Defines the mapping of two particular ClientServerOperations in context of two different ClientServer Interfaces.			
Base	ARObject			
Aggregated by	ClientServerInterfaceMapping.operationMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
argument Mapping	DataPrototypeMapping	*	aggr	Defines the mapping of two particular ArgumentData Prototypes with unequal names or unequal semantic (resolution or range) in context of Operations.
firstOperation	ClientServerOperation	0..1	ref	First to-be-mapped ClientServerOperation of a Client ServerInterface.
firstToSecond Data Transformation	DataTransformation	0..1	ref	This reference indicates that a DataTransformation is intended in the context of the ClientServerOperation Mapping.
second Operation	ClientServerOperation	0..1	ref	Second to-be-mapped ClientServerOperation of a Client ServerInterface.

Table A.176: ClientServerOperationMapping

Class	ClientServerToSignalMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	This element maps the ClientServerOperation to call- and return-SystemSignals.			
Base	ARObject, DataMapping			
Aggregated by	SystemMapping.dataMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
callSignal	SystemSignal	1	ref	Reference to the callSignal to which the IN and INOUT ArgumentDataPrototypes are mapped.
clientServer Operation	ClientServerOperation	1	iref	Reference to a ClientServerOperation, which is mapped to a call SystemSignal and a return SystemSignal. **InstanceRef implemented by:**OperationInSystem InstanceRef
returnSignal	SystemSignal	0..1	ref	Reference to the returnSignal to which the OUT and INOUT ArgumentDataPrototypes are mapped.

Table A.177: ClientServerToSignalMapping

Class	Code			
Package	M2::AUTOSARTemplates::CommonStructure::Implementation			
Note	A generic code descriptor. The type of the code (source or object) is defined via the category attribute of the associated engineering object.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*Implementation*.codeDescriptor			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
artifact Descriptor	AutosarEngineering Object	*	aggr	Refers to the artifact belonging to this code descriptor.
callbackHeader	ServiceNeeds	*	ref	The association callbackHeader describes in which header files the function declarations of callback functions are provided to a service module. With this information the service module can include the appropriate header files in its configuration files.

Table A.178: Code

Class	Collection			
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::ElementCollection			
Note	This meta-class specifies a collection of elements. A collection can be utilized to express additional aspects for a set of elements. Note that Collection is an ARElement. Therefore it is applicable e.g. for EvaluatedVariant, even if this is not obvious. Usually the category of a Collection is "SET". On the other hand, a Collection can also express an arbitrary relationship between elements. This is denoted by the category "RELATION" (see also [TPS_GST_00347]). In this case the collection represents an association from "sourceElement" to "targetElement" in the role "role". **Tags:**atp.recommendedPackage=Collections			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
autoCollect	AutoCollectEnum	0..1	attr	This attribute reflects how far the referenced objects are part of the collection. **Tags:**xml.sequenceOffset=20
collected Instance	AtpFeature	*	iref	This instance ref supports the use case that a particular instance is part of the collection. **Tags:**xml.sequenceOffset=60 **InstanceRef implemented by:**AnyInstanceRef
collection Semantics	NameToken	0..1	attr	Provides the ability to express the semantics of a Collection depending on the intended use case. The collectionSemantics is specified as a NameToken which must be agreed by all stakeholders. **Tags:**xml.sequenceOffset=25
element	Identifiable	*	ref	This is an element in the collection. Note that Collection itself is collectable. Therefore collections can be nested. In case of category="RELATION" this represents the target end of the relation. **Tags:**xml.sequenceOffset=40

▽

△

Class	Collection			
elementRole	Identifier	0..1	attr	This attribute allows to denote a particular role of the collection. Note that the applicable semantics shall be mutually agreed between the two parties. In particular it denotes the role of element in the context of sourceElement. **Tags:**xml.sequenceOffset=30
sourceElement	Identifiable	*	ref	Only if Category = "RELATION". This represents the source of a relation. **Tags:**xml.sequenceOffset=50
sourceInstance	AtpFeature	*	iref	Only if Category = "RELATION". This represents the source instance of a relation. **Tags:**xml.sequenceOffset=70 **InstanceRef implemented by:**AnyInstanceRef

Table A.179: Collection

Class	*CommConnectorPort* (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology			
Note	The Ecu communication relationship defines which signals, Pdus and frames are actually received and transmitted by this ECU. For each signal, Pdu or Frame that is transmitted or received and used by the Ecu an association between an ISignalPort, IPduPort or FramePort with the corresponding Triggering shall be created. An ISignalPort shall be created only if the corresponding signal is handled by COM (RTE or Signal Gateway). If a Pdu Gateway ECU only routes the Pdu without being interested in the content only a FramePort and an IPduPort needs to be created.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	FramePort, IPduPort, ISignalPort			
Aggregated by	*CommunicationConnector*.ecuCommPortInstance			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
communication Direction	Communication DirectionType	1	attr	Communication Direction of the Connector Port (input or output Port).

Table A.180: CommConnectorPort

Class	**CommunicationBufferLocking**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::PortAPIOptions			
Note	The aggregation of this meta-class specifies that a RunnableEntity supports locked communication buffers supplied by the RTE. It is able to cope with the error RTE_E_COM_BUSY.			
Base	*ARObject*, *SwcSupportedFeature*			
Aggregated by	PortAPIOption.supportedFeature			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
supportBuffer Locking	SupportBufferLocking Enum	0..1	attr	This attribute is used to indicate the intended buffer locking behavior.

Table A.181: CommunicationBufferLocking

Class	<<atpVariation>> *CommunicationCluster* (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology			
Note	The CommunicationCluster is the main element to describe the topological connection of communicating ECUs. A cluster describes the ensemble of ECUs, which are linked by a communication medium of arbitrary topology (bus, star, ring, ...). The nodes within the cluster share the same communication protocol, which may be event-triggered, time-triggered or a combination of both. A CommunicationCluster aggregates one or more physical channels. **Tags:**vh.latestBindingTime=postBuild			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Subclasses	*AbstractCanCluster*, EthernetCluster, FlexrayCluster, LinCluster, UserDefinedCluster			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
baudrate	PositiveUnlimitedInteger	0..1	attr	Channels speed in bits/s.
physical Channel	PhysicalChannel	1..*	aggr	This relationship defines which channel element belongs to which cluster. A channel shall be assigned to exactly one cluster, whereas a cluster may have one or more channels. Note: This atpSplitable property has no atp.Splitkey due to atpVariation (PropertySetPattern). **Stereotypes:** atpSplitable; atpVariation **Tags:**vh.latestBindingTime=systemDesignTime
protocolName	String	0..1	attr	The name of the protocol used.
protocolVersion	String	0..1	attr	The version of the protocol used.

Table A.182: CommunicationCluster

Class	*CommunicationConnector* (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology			
Note	The connection between the referencing ECU and the referenced channel via the referenced controller. Connectors are used to describe the bus interfaces of the ECUs and to specify the sending/receiving behavior. Each CommunicationConnector has a reference to exactly one communicationController. Note: Several CommunicationConnectors can be assigned to one PhysicalChannel in the scope of one ECU Instance.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	*AbstractCanCommunicationConnector*, EthernetCommunicationConnector, FlexrayCommunication Connector, LinCommunicationConnector, UserDefinedCommunicationConnector			
Aggregated by	EcuInstance.connector, MachineDesign.communicationConnector			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
commController	Communication Controller	1	ref	Reference to the communication controller. The CommunicationConnector and referenced CommunicationController shall be aggregated by the same ECUInstance. The communicationController can be referenced by several CommunicationConnector elements. This is important for the FlexRay Bus. FlexRay communicates via two physical channels. But only one controller in an ECU is responsible for both channels. Thus, two connectors (for channel A and for channel B) shall reference to the same controller.

▽

△

Class	CommunicationConnector (abstract)			
createEcu WakeupSource	Boolean	0..1	attr	If this parameter is available and set to true then a channel wakeup source shall be created for the Physical Channel referencing this CommunicationConnector.
dynamicPncTo Channel Mapping Enabled	Boolean	0..1	attr	Defines if this EcuInstance shall implement the dynamic PNC-to-channel-mapping functionality on this CommunicationConnector and its respective Physical Channel. **Tags:**atp.Status=draft
ecuCommPort Instance	CommConnectorPort	*	aggr	An ECUs reception or send ports. atpVariation: If signals/PDUs/frames are variable, the corresponding ports shall be variable, too. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=ecuCommPortInstance.shortName, ecu CommPortInstance.variationPoint.shortLabel vh.latestBindingTime=postBuild
pncFilterArray Mask (ordered)	PositiveInteger	*	attr	Bit mask for NM-Pdu Payload used to configure the NM filter mask for the Network Management. **Tags:**atp.Status=draft
pncGateway Type	PncGatewayTypeEnum	0..1	attr	Defines if this EcuInstance shall implement the Pnc Gateway functionality on this CommunicationConnector and its respective PhysicalChannel. Several Ecu Instances on the same PhysicalChannel can have the PncGateway functionality enabled, but only one of them shall have the pncGatewayType "active".

Table A.183: CommunicationConnector

Class	<<atpVariation>> CommunicationController (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology			
Note	The communication controller is a dedicated hardware device by means of which hosts are sending frames to and receiving frames from the communication medium. **Tags:**vh.latestBindingTime=postBuild			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Subclasses	AbstractCanCommunicationController, EthernetCommunicationController, FlexrayCommunication Controller, LinCommunicationController, UserDefinedCommunicationController			
Aggregated by	EcuInstance.commController, MachineDesign.communicationController			
Attribute	Type	Mult.	Kind	Note
wakeUpBy Controller Supported	Boolean	0..1	attr	Defines whether the ECU shall be woken up by this CommunicationController. TRUE: wake up is possible FALSE: wake up is not supported Note: If wakeUpByControllerSupported is set to TRUE the feature shall be supported by both hardware and basic software.

Table A.184: CommunicationController

Class	CommunicationControllerMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::ECUResourceMapping			
Note	CommunicationControllerMapping specifies the CommunicationPeripheral hardware (defined in the ECU Resource Template) to realize the specified CommunicationController in a physical topology.			
Base	ARObject			
Aggregated by	ECUMapping.commControllerMapping			
Attribute	Type	Mult.	Kind	Note
communication Controller	Communication Controller	1	ref	Reference to the CommunicationController in the System Template
hw Communication Controller	HwElement	1	ref	Reference to a HwElement of category Communication Controller in the ECU Resource Template.

Table A.185: CommunicationControllerMapping

Enumeration	CommunicationDirectionType
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication
Note	Describes the communication direction.
Aggregated by	CommConnectorPort.communicationDirection, IPSecRule.direction, ISignalIPduGroup.communicationDirection
Literal	Description
in	Reception (Input) **Tags:**atp.EnumerationLiteralIndex=0
out	Transmission (Output) **Tags:**atp.EnumerationLiteralIndex=1

Table A.186: CommunicationDirectionType

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, Atp Type, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable, Sw ComponentType			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
hardware Element	HwDescriptionEntity	*	ref	Reference from the ComplexDeviceDriverSwComponent Type to the description of the used HwElements.

Table A.187: ComplexDeviceDriverSwComponentType

Class	ComponentClustering
Package	M2::AUTOSARTemplates::SystemTemplate::SWmapping
Note	Constraint that forces the mapping of all referenced SW component instances to the same ECU, Core, Partition depending on the defined mappingScope attribute. If mappingScope is not specified then mappingScopeEcu shall be assumed.

▽

△

Class	ComponentClustering			
Base	*ARObject*, *MappingConstraint*			
Aggregated by	SystemMapping.mappingConstraint			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
clustered Component	SwComponent Prototype	1..*	iref	Reference to the components that have to be mapped together. **InstanceRef implemented by:**ComponentInSystem InstanceRef
mappingScope	MappingScopeEnum	0..1	attr	This attribute indicates whether the ComponentClustering mapping constraint applies to different ECUs, partitions or cores. If this attribute is not specified then mappingScope Ecu shall be assumed.

Table A.188: ComponentClustering

Class	ComponentSeparation			
Package	M2::AUTOSARTemplates::SystemTemplate::SWmapping			
Note	Constraint that forces the two referenced SW components (called A and B in the following) not to be mapped to the same ECU, Core, Partition depending on the defined mappingScope attribute. If mapping Scope is not specified then mappingScopeEcu shall be assumed. If a SW component (e.g. A) is a composition, none of the atomic SW components making up the A composition shall be mapped together with any of the atomic SW components making up the B composition. Furthermore, A and B shall be disjoint.			
Base	*ARObject*, *MappingConstraint*			
Aggregated by	SystemMapping.mappingConstraint			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
mappingScope	MappingScopeEnum	0..1	attr	This attribute indicates whether the Component Separation mapping constraint applies to different ECUs, partitions or cores. If this attribute is not specified then mappingScopeEcu shall be assumed.
separated Component	SwComponent Prototype	2	iref	The two components that have to be mapped to different ECUs **InstanceRef implemented by:**ComponentInSystem InstanceRef

Table A.189: ComponentSeparation

Class	CompositeNetworkRepresentation			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	This meta-class is used to define the network representation of leaf elements of composite application data types.			
Base	*ARObject*			
Aggregated by	*ReceiverComSpec*.compositeNetworkRepresentation, *SenderComSpec*.compositeNetwork Representation			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
leafElement	ApplicationComposite ElementDataPrototype	0..1	iref	This represents that leaf element of an application composite data type. **InstanceRef implemented by:**ApplicationComposite ElementInPortInterfaceInstanceRef

▽

△

Class	CompositeNetworkRepresentation			
network Representation	SwDataDefProps	0..1	aggr	The SwDataDefProps owned by the CompositeNetwork Representation are used to define the network representation of the leaf element of an Application CompositeDataType. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=networkRepresentation

Table A.190: CompositeNetworkRepresentation

Class	CompositeRuleBasedValueSpecification			
Package	M2::AUTOSARTemplates::CommonStructure::Constants			
Note	This meta-class represents rule-based values for DataPrototypes typed by composite AutosarDataTypes.			
Base	ARObject, AbstractRuleBasedValueSpecification, ValueSpecification			
Aggregated by	ApplicationAssocMapElementValueSpecification.key, ApplicationAssocMapElementValueSpecification. value, ArrayValueSpecification.element, CalibrationParameterValue.applInitValue, CalibrationParameter Value.implInitValue, ConstantSpecification.valueSpec, CryptoServiceKey.developmentValue, Diagnostic EnvDataCondition.compareValue, DiagnosticEnvDataElementCondition.compareValue, FieldSenderCom Spec.initValue, ISignal.initValue, ISignal.timeoutSubstitutionValue, NonqueuedReceiverComSpec.init Value, NonqueuedReceiverComSpec.timeoutSubstitutionValue, NonqueuedSenderComSpec.initValue, NvProvideComSpec.ramBlockInitValue, NvProvideComSpec.romBlockInitValue, NvRequireComSpec.init Value, ParameterDataPrototype.initValue, ParameterProvideComSpec.initValue, ParameterRequireCom Spec.initValue, PersistencyDataRequiredComSpec.initValue, PersistencyKeyValuePair.initValue, Port DefinedArgumentValue.value, PortPrototypeBlueprintInitValue.value, RecordValueSpecification.field, StateManagementCompareCondition.compareValue, SwDataDefProps.invalidValue, VariableData Prototype.initValue			
Attribute	Type	Mult.	Kind	Note
argument (ordered)	CompositeValue Specification	*	aggr	This represents the collection of aggregated Value Specifications. The last ValueSpecification in the collection shall be taken to execute the filling rule. **Tags:**xml.sequenceOffset=30
compound Primitive Argument (ordered)	CompositeRuleBased ValueArgument	*	aggr	This represents the collection of aggregated Value Specifications for compound primitive data type. The last ValueSpecification in the collection shall be taken to execute the filling rule. **Tags:**xml.sequenceOffset=35
maxSizeToFill	PositiveInteger	0..1	attr	If a rule is chosen which does not fill until the end, this determines until which size the rule shall fill the values. **Tags:**xml.sequenceOffset=40
rule	Identifier	0..1	attr	This denotes the name of the rule of the RuleBasedValue Specification. The rule determines the calculation specification according which the arguments are used to calculated the values. **Tags:**xml.sequenceOffset=20

Table A.191: CompositeRuleBasedValueSpecification

Class	CompositeValueSpecification (abstract)
Package	M2::AUTOSARTemplates::CommonStructure::Constants
Note	This abstract meta-class acts a base class for ValueSpecifications that have a composite form.
Base	ARObject, ValueSpecification
Subclasses	ArrayValueSpecification, RecordValueSpecification

▽

△

Class	**CompositeValueSpecification** (abstract)			
Aggregated by	ApplicationAssocMapElementValueSpecification.key, ApplicationAssocMapElementValueSpecification. value, ArrayValueSpecification.element, CalibrationParameterValue.applInitValue, CalibrationParameter Value.implInitValue, CompositeRuleBasedValueSpecification.argument, ConstantSpecification.value Spec, CryptoServiceKey.developmentValue, DiagnosticEnvDataCondition.compareValue, DiagnosticEnv DataElementCondition.compareValue, FieldSenderComSpec.initValue, ISignal.initValue, ISignal.timeout SubstitutionValue, NonqueuedReceiverComSpec.initValue, NonqueuedReceiverComSpec.timeout SubstitutionValue, NonqueuedSenderComSpec.initValue, NvProvideComSpec.ramBlockInitValue, Nv ProvideComSpec.romBlockInitValue, NvRequireComSpec.initValue, ParameterDataPrototype.initValue, ParameterProvideComSpec.initValue, ParameterRequireComSpec.initValue, PersistencyDataRequired ComSpec.initValue, PersistencyKeyValuePair.initValue, PortDefinedArgumentValue.value, PortPrototype BlueprintInitValue.value, RecordValueSpecification.field, *StateManagementCompareCondition*.compare Value, SwDataDefProps.invalidValue, VariableDataPrototype.initValue			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.192: CompositeValueSpecification

Class	**CompositionSwComponentType**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Composition			
Note	A CompositionSwComponentType aggregates SwComponentPrototypes (that in turn are typed by Sw ComponentTypes) as well as SwConnectors for primarily connecting SwComponentPrototypes among each others and towards the surface of the CompositionSwComponentType. By this means, hierarchical structures of software-components can be created. **Tags:**atp.recommendedPackage=SwComponentTypes			
Base	*ARElement*, ARObject, *AtpBlueprint*, AtpBlueprintable, *AtpClassifier*, *AtpType*, CollectableElement, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SwComponentType*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
component	SwComponent Prototype	*	aggr	The instantiated components that are part of this composition. The aggregation of SwComponentPrototype is subject to variability with the purpose to support the conditional existence of a SwComponentPrototype. Please be aware: if the conditional existence of Sw ComponentPrototypes is resolved post-build the deselected SwComponentPrototypes are still contained in the ECUs build but the instances are inactive in that they are not scheduled by the RTE. The aggregation is marked as atpSplitable in order to allow the addition of service components to the ECU extract during the ECU integration. The use case for having 0 components owned by the CompositionSwComponentType could be to deliver an empty CompositionSwComponentType to e.g. a supplier for filling the internal structure. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=component.shortName, component.variation Point.shortLabel vh.latestBindingTime=postBuild

▽

△

Class	CompositionSwComponentType			
connector	SwConnector	*	aggr	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. The aggregation of SwConnectors is subject to variability with the purpose to support variant data flow. The aggregation is marked as atpSplitable in order to allow the extension of the ECU extract with AssemblySw Connectors between ApplicationSwComponentTypes and ServiceSwComponentTypes during the ECU integration. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=connector.shortName, connector.variation Point.shortLabel vh.latestBindingTime=postBuild
constantValue Mapping	ConstantSpecification MappingSet	*	ref	Reference to the ConstantSpecificationMapping to be applied for initValues of PPortComSpecs and RPortCom Spec. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=constantValueMapping
dataType Mapping	DataTypeMappingSet	*	ref	Reference to the DataTypeMapping to be applied for the used ApplicationDataTypes in PortInterfaces. 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 CompositionSwComponent Types. Rationale: if the mapping of ApplicationDataTypes on the delegated and inner PortPrototype matches then the mapping to ImplementationDataTypes is not impacting compatibility. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=dataTypeMapping
instantiation RTEEventProps	InstantiationRTEEvent Props	*	aggr	This allows to define instantiation specific properties for RTE Events, in particular for instance specific scheduling. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=instantiationRTEEventProps.shortLabel, instantiationRTEEventProps.variationPoint.shortLabel vh.latestBindingTime=codeGenerationTime

Table A.193: CompositionSwComponentType

Class	Compu			
Package	M2::MSR::AsamHdo::ComputationMethod			
Note	This meta-class represents the ability to express one particular computation.			
Base	ARObject			
Aggregated by	CompuMethod.compuInternalToPhys, CompuMethod.compuPhysToInternal			
Attribute	Type	Mult.	Kind	Note

▽

△

Class	Compu			
compuContent	CompuContent	0..1	aggr	This specifies the details of the computation. **Tags:** xml.roleElement=false xml.roleWrapperElement=false xml.sequenceOffset=20 xml.typeElement=false xml.typeWrapperElement=false
compuDefault Value	CompuConst	0..1	aggr	This property can be used to specify an output value for a conversion formula, if the value to be converted lies outside the plausibility limit. Although this is possible for all conversion formulae, it is especially valid for variables with tabular conversion formulae. **Tags:**xml.sequenceOffset=70

Table A.194: Compu

Class	CompuConst			
Package	M2::MSR::AsamHdo::ComputationMethod			
Note	This meta-class represents the fact that the value of a computation method scale is constant.			
Base	ARObject			
Aggregated by	Compu.compuDefaultValue, CompuScale.compuInverseValue, CompuScaleConstantContents.compuConst			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
compuConst ContentType	CompuConstContent	0..1	aggr	This is the actual content of the constant compu method scale. **Tags:** xml.roleElement=false xml.roleWrapperElement=false xml.sequenceOffset=10 xml.typeElement=false xml.typeWrapperElement=false

Table A.195: CompuConst

Class	CompuConstFormulaContent			
Package	M2::MSR::AsamHdo::ComputationMethod			
Note	This meta-class represents the fact that the constant value of the computation method is represented by a variation point. This difference is due to compatibility with ASAM HDO.			
Base	ARObject, CompuConstContent			
Aggregated by	CompuConst.compuConstContentType			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	CompuConstFormulaContent			
vf	Numerical	1	attr	Value calculated via a system constant. This element is included in every case where parameters should be generated from numerical values during compile time (not runtime!). Thus for example, the influence of the cylinder number on conversion formulae can be introduced in a repeatable manner. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=codeGenerationTime xml.sequenceOffset=30

Table A.196: CompuConstFormulaContent

Class	CompuConstTextContent			
Package	M2::MSR::AsamHdo::ComputationMethod			
Note	This meta-class represents the textual content of a scale.			
Base	*ARObject*, *CompuConstContent*			
Aggregated by	CompuConst.compuConstContentType			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
vt	VerbatimString	0..1	attr	This represents a textual constant in the computation method.

Table A.197: CompuConstTextContent

Class	CompuMethod			
Package	M2::MSR::AsamHdo::ComputationMethod			
Note	This meta-class represents the ability to express the relationship between a physical value and the mathematical representation. Note that this is still independent of the technical implementation in data types. It only specifies the formula how the internal value corresponds to its physical pendant. **Tags:**atp.recommendedPackage=CompuMethods			
Base	*ARElement*, ARObject, *AtpBlueprint*, AtpBlueprintable, CollectableElement, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
compuInternal ToPhys	Compu	0..1	aggr	This specifies the computation from internal values to physical values. **Tags:**xml.sequenceOffset=80
compuPhysTo Internal	Compu	0..1	aggr	This represents the computation from physical values to the internal values. **Tags:**xml.sequenceOffset=90
displayFormat	DisplayFormatString	0..1	attr	This property specifies, how the physical value shall be displayed e.g. in documents or measurement and calibration tools. **Tags:**xml.sequenceOffset=20
unit	Unit	0..1	ref	This is the physical unit of the Physical values for which the CompuMethod applies. **Tags:**xml.sequenceOffset=30

Table A.198: CompuMethod

Class	CompuNominatorDenominator			
Package	M2::MSR::AsamHdo::ComputationMethod			
Note	This class represents the ability to express a polynomial either as Nominator or as Denominator.			
Base	ARObject			
Aggregated by	CompuRationalCoeffs.compuDenominator, CompuRationalCoeffs.compuNumerator			
Attribute	Type	Mult.	Kind	Note
v (ordered)	Numerical	*	attr	this is the list of polynomial factors. Note that the first vf represents the power=0. The polynomial is v[0] * x^0 + v[1] * x^1 ... **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.roleElement=true xml.roleWrapperElement=false xml.sequenceOffset=20 xml.typeElement=false xml.typeWrapperElement=false

Table A.199: CompuNominatorDenominator

Class	CompuRationalCoeffs			
Package	M2::MSR::AsamHdo::ComputationMethod			
Note	This meta-class represents the ability to express a rational function by specifying the coefficients of nominator and denominator.			
Base	ARObject			
Aggregated by	CompuScaleRationalFormula.compuRationalCoeffs			
Attribute	Type	Mult.	Kind	Note
compuDenominator	CompuNominatorDenominator	0..1	aggr	This is the denominator of the expression. **Tags:**xml.sequenceOffset=30
compuNumerator	CompuNominatorDenominator	0..1	aggr	This is the numerator of the rational expression. **Tags:**xml.sequenceOffset=20

Table A.200: CompuRationalCoeffs

Class	CompuScale			
Package	M2::MSR::AsamHdo::ComputationMethod			
Note	This meta-class represents the ability to specify one segment of a segmented computation method.			
Base	ARObject			
Aggregated by	CompuScales.compuScale			
Attribute	Type	Mult.	Kind	Note
compuInverseValue	CompuConst	0..1	aggr	This is the inverse value of the constraint. This supports the case that the scale is not reversible per se. **Tags:**xml.sequenceOffset=60
compuScaleContents	CompuScaleContents	0..1	aggr	This represents the computation details of the scale. **Tags:** xml.roleElement=false xml.roleWrapperElement=false xml.sequenceOffset=70 xml.typeElement=false xml.typeWrapperElement=false

▽

△

Class	CompuScale			
desc	MultiLanguageOverview Paragraph	0..1	aggr	\<desc\> represents a general but brief description of the object in question. **Tags:**xml.sequenceOffset=30
lowerLimit	Limit	0..1	attr	This specifies the lower limit of the scale. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=40
mask	PositiveUnlimitedInteger	0..1	attr	In difference to all the other computational methods every COMPU-SCALE will be applied including the bit MASK. Therefore it is allowed for this type of COMPU-METHOD, that COMPU-SCALES overlap. To calculate the string reverse to a value, the string has to be split and the according value for each substring has to be summed up. The sum is finally transmitted. The processing has to be done in order of the COMPU-SCALE elements. **Tags:**xml.sequenceOffset=35
shortLabel	Identifier	0..1	attr	This element specifies a short name for the particular scale. The name can for example be used to derive a programming language identifier. **Tags:**xml.sequenceOffset=20
symbol	CIdentifier	0..1	attr	The symbol, if provided, is used by code generators to get a C identifier for the CompuScale. The name will be used as is for the code generation, therefore it needs to be unique within the generation context. **Tags:**xml.sequenceOffset=25
upperLimit	Limit	0..1	attr	This specifies the upper limit of a of the scale. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=50

Table A.201: CompuScale

Class	CompuScaleConstantContents			
Package	M2::MSR::AsamHdo::ComputationMethod			
Note	This meta-class represents the fact that a particular scale of the computation method is constant.			
Base	*ARObject*, *CompuScaleContents*			
Aggregated by	CompuScale.compuScaleContents			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
compuConst	CompuConst	0..1	aggr	This represents the fact that the scale is a constant. The use case is mainly a non interpolated scale. It is a simplification of the fact that a constant scale can also be expressed as rational function of order 0. **Tags:**xml.sequenceOffset=90

Table A.202: CompuScaleConstantContents

Class	CompuScales			
Package	M2::MSR::AsamHdo::ComputationMethod			
Note	This meta-class represents the ability to stepwise express a computation method.			
Base	*ARObject*, *CompuContent*			
Aggregated by	Compu.compuContent			
Attribute	Type	Mult.	Kind	Note
compuScale (ordered)	CompuScale	*	aggr	This represents one scale within the compu method. Note that it contains a Variationpoint in order to support blueprints of enumerations. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=compuScale, compuScale.variation Point.shortLabel vh.latestBindingTime=blueprintDerivationTime xml.roleElement=true xml.roleWrapperElement=true xml.sequenceOffset=40 xml.typeElement=false xml.typeWrapperElement=false

Table A.203: CompuScales

Class	ConcreteClassTailoring			
Package	M2::AUTOSARTemplates::CommonStructure::StandardizationTemplate::DataExchangePoint::Data FormatTailoring			
Note	Tailoring of concrete meta classes.			
Base	*ARObject*, *ClassTailoring*, *DataFormatElementReference*, *DataFormatElementScope*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *SpecElementReference*, *SpecElementScope*			
Aggregated by	AggregationTailoring.typeTailoring, DataFormatTailoring.classTailoring, ReferenceTailoring.typeTailoring			
Attribute	Type	Mult.	Kind	Note
validationRoot	Boolean	0..1	attr	Specification if this concrete Meta-Class is a root element for validation. I.e.: The validation starts at an object of this concrete Meta-Class and continues by following all aggregations and references that are in scope of this Data Exchange Point. **Tags:**xml.sequenceOffset=10

Table A.204: ConcreteClassTailoring

Class	ConcretePatternEventTriggering			
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint::EventTriggeringConstraint			
Note	The ConcretePatternEventTriggering describes the behavior of an event, which occurs following a precisely known pattern.			
Base	*ARObject*, *EventTriggeringConstraint*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *Timing Constraint*, *Traceable*			
Aggregated by	*TimingExtension*.timingGuarantee, *TimingExtension*.timingRequirement			
Attribute	Type	Mult.	Kind	Note

$\bigtriangledown$

Document ID 635: AUTOSAR_TR_ClassicAutosarModelConstraints

△

Class	ConcretePatternEventTriggering			
offset	MultidimensionalTime	*	aggr	The offset for each occurrence of the event in the specified time interval. **Tags:** xml.name=TIME-VALUE xml.roleElement=true xml.sequenceOffset=10 xml.typeElement=false
patternJitter	MultidimensionalTime	0..1	aggr	The optional parameter "Pattern Jitter" specifies the 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 "Pattern Period".
patternLength	MultidimensionalTime	0..1	aggr	The length of the observed time interval. **Tags:**xml.sequenceOffset=20
patternPeriod	MultidimensionalTime	0..1	aggr	The optional parameter "Pattern Period" specifies the time distance between the beginnings of subsequent repetitions of the given concrete pattern.

Table A.205: ConcretePatternEventTriggering

Class	<<atpMixedString>> **ConditionByFormula**			
Package	M2::AUTOSARTemplates::GenericStructure::VariantHandling			
Note	This class represents a condition which is computed based on system constants according to the specified expression. The expected result is considered as boolean value. The result of the expression is interpreted as a condition. • "0" represents "false"; • a value other than zero is considered "true"			
Base	*ARObject*, *FormulaExpression*, *SwSystemconstDependentFormula*			
Aggregated by	VariationPoint.swSyscond, VariationPointProxy.conditionAccess			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
bindingTime	BindingTimeEnum	1	attr	This attribute specifies the point in time when condition may be evaluated at earliest. At this point in time all referenced system constants shall have a value. **Tags:**xml.attribute=true

Table A.206: ConditionByFormula

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	*ARObject*			
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.
propability	Float	0..1	attr	The probability for the measured lower and upper bound of the confidence interval.
upperBound	MultidimensionalTime	0..1	aggr	The upper bound of the expected distance of two consecutive event occurrences.

Table A.207: ConfidenceInterval

Class	ConsistencyNeeds			
Package	M2::AUTOSARTemplates::SWComponentTemplate::ImplicitCommunicationBehavior			
Note	This meta-class represents the ability to define requirements on the implicit communication behavior.			
Base	*ARObject*, *AtpBlueprint*, *AtpBlueprintable*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ConsistencyNeedsBlueprintSet.consistencyNeeds, *SwComponentType*.consistencyNeeds			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dpgDoesNot Require Coherency	DataPrototypeGroup	*	aggr	This group of VariableDataPrototypes does not require coherency with respect to the implicit communication behavior. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=dpgDoesNotRequireCoherency.shortName, dpgDoesNotRequireCoherency.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
dpgRequires Coherency	DataPrototypeGroup	*	aggr	This group of VariableDataPrototypes requires coherency with respect to the implicit communication behavior, i.e. all read and write access to VariableDataPrototypes in the DataPrototypeGroup by the RunnableEntitys of the RunnableEntityGroup need to be handled in a coherent manner. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=dpgRequiresCoherency.shortName, dpg RequiresCoherency.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
regDoesNot RequireStability	RunnableEntityGroup	*	aggr	This group of RunnableEntities does not require stability with respect to the implicit communication behavior. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=regDoesNotRequireStability.shortName, reg DoesNotRequireStability.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
regRequires Stability	RunnableEntityGroup	*	aggr	This group of RunnableEntities requires stability with respect to the implicit communication behavior, i.e. all read and write access to VariableDataPrototypes in the DataPrototypeGroup by the RunnableEntitys of the RunnableEntityGroup need to be handled in a stable manner. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=regRequiresStability.shortName, reg RequiresStability.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table A.208: ConsistencyNeeds

Class	ConstantReference
Package	M2::AUTOSARTemplates::CommonStructure::Constants
Note	Instead of defining this value inline, a constant is referenced.
Base	*ARObject*, *ValueSpecification*

▽

△

Class	ConstantReference			
Aggregated by	ApplicationAssocMapElementValueSpecification.key, ApplicationAssocMapElementValueSpecification.value, ArrayValueSpecification.element, CalibrationParameterValue.applInitValue, CalibrationParameterValue.implInitValue, ConstantSpecification.valueSpec, CryptoServiceKey.developmentValue, DiagnosticEnvDataCondition.compareValue, DiagnosticEnvDataElementCondition.compareValue, FieldSenderComSpec.initValue, ISignal.initValue, ISignal.timeoutSubstitutionValue, NonqueuedReceiverComSpec.initValue, NonqueuedReceiverComSpec.timeoutSubstitutionValue, NonqueuedSenderComSpec.initValue, NvProvideComSpec.ramBlockInitValue, NvProvideComSpec.romBlockInitValue, NvRequireComSpec.initValue, ParameterDataPrototype.initValue, ParameterProvideComSpec.initValue, ParameterRequireComSpec.initValue, PersistencyDataRequiredComSpec.initValue, PersistencyKeyValuePair.initValue, PortDefinedArgumentValue.value, PortPrototypeBlueprintInitValue.value, RecordValueSpecification.field, StateManagementCompareCondition.compareValue, SwDataDefProps.invalidValue, VariableDataPrototype.initValue			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
constant	ConstantSpecification	0..1	ref	The referenced constant.

Table A.209: ConstantReference

Class	ConstantSpecification			
Package	M2::AUTOSARTemplates::CommonStructure::Constants			
Note	Specification of a constant that can be part of a package, i.e. it can be defined stand-alone. **Tags:**atp.recommendedPackage=ConstantSpecifications			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
valueSpec	ValueSpecification	0..1	aggr	Specification of an expression leading to a value for this constant.

Table A.210: ConstantSpecification

Class	ConstantSpecificationMapping			
Package	M2::AUTOSARTemplates::CommonStructure::Constants			
Note	This meta-class is used to create an association of two ConstantSpecifications. One ConstantSpecification is supposed to be defined in the application domain while the other should be defined in the implementation domain. Hence the ConstantSpecificationMapping needs to be used where a ConstantSpecification defined in one domain needs to be associated to a ConstantSpecification in the other domain. This information is crucial for the RTE generator.			
Base	*ARObject*			
Aggregated by	ConstantSpecificationMappingSet.mapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
applConstant	ConstantSpecification	0..1	ref	A ConstantSpecification defined in the application domain.
implConstant	ConstantSpecification	0..1	ref	A ConstantSpecification defined in the implementation domain.

Table A.211: ConstantSpecificationMapping

Class	ConsumedEventGroup			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances			
Note	This element represents an event-group to which the service consumer wants to subscribe.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ConsumedServiceInstance.consumedEventGroup			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
application Endpoint	ApplicationEndpoint	0..1	ref	Defines the application endpoint where the events of the event group are received in case of multicast reception. **Tags:**atp.Status=obsolete
autoRequire	Boolean	0..1	attr	Defines that this ConsumedEventGroup shall be requested (subscribed) as soon as the corresponding ConsumedServiceInstance is requested. This could be at ECU start, if ConsumedServiceInstance.autoRequire is set to TRUE or as soon as the ConsumedServiceInstance is requested by the application, if ConsumedService Instance.autoRequire is set to FALSE.
eventGroup Identifier	PositiveInteger	0..1	attr	EventGroup ID. Shall be unique within one system to allow service discovery.
eventMulticast Address	ApplicationEndpoint	*	ref	This reference defines the multicast address or a multicast address resource where the events of the event group are received. If the multicast address is determined via configuration and not at runtime via service discovery this reference points to the multicast address over which the events will be received. If the multicast address is determined at runtime via service discovery this reference shall be used to define the necessary local multicast address resources, i.e. RAM space in the TcpIp module in which the multicast address is stored at runtime. Please note that in this case the referenced address may be defined as ANY UDP port and ANY IP address since the multicast address will be received at runtime. If several multicast addresses are considered to be used the ConsumedEventGroup shall point to different ApplicationEndpoint objects to reserve the necessary resources in the configuration. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=eventMulticastAddress.applicationEndpoint, eventMulticastAddress.variationPoint.shortLabel vh.latestBindingTime=postBuild
pduActivation RoutingGroup	PduActivationRouting Group	*	aggr	The ServiceDiscovery module is able to activate and deactivate the PDU routing for receiving events.
priority	PositiveInteger	0..1	attr	Defines the frame priority where values from 0 (best effort) to 7 (highest) are allowed.
routingGroup	SoAdRoutingGroup	*	ref	The ServiceDiscovery module is able to activate and deactivate the PDU routing for receiving events. **Tags:**atp.Status=obsolete
sdClientConfig	SdClientConfig	0..1	aggr	The readiness to receive events is defined by the Service Discovery of the ConsumedEventGroup. The Event Handler shall know about this announcement to decide about the submission of events. Therefore the Event Handler may be configured with Service-Discovery Client attributes. **Tags:**atp.Status=obsolete

▽

   Document ID 635: AUTOSAR_TR_ClassicAutosarModelConstraints

△

Class	ConsumedEventGroup			
sdClientTimer Config	SomeipSdClientEvent GroupTimingConfig	0..1	ref	Client Timing configuration settings that are EventGroup specific. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=sdClientTimerConfig.someipSdClientEvent GroupTimingConfig, sdClientTimerConfig.variation Point.shortLabel vh.latestBindingTime=postBuild

Table A.212: ConsumedEventGroup

Class	ConsumedServiceInstance			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances			
Note	Service instances that are consumed by the ECU that is connected via the ApplicationEndpoint to a CommunicationConnector.			
Base	ARObject, AbstractServiceInstance, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	ApplicationEndpoint.consumedServiceInstance, ServiceInstanceCollectionSet.serviceInstance			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
autoRequire	Boolean	0..1	attr	Defines that this ConsumedServiceInstance shall be required (searched for) by the service discovery at ECU start.
blocklisted Version	SomeipServiceVersion	*	aggr	Collection of blocklisted versions **Tags:**atp.Status=draft
consumedEvent Group	ConsumedEventGroup	*	aggr	Selection of event-groups the consumer wants to subscribe for. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=consumedEventGroup.shortName, consumedEventGroup.variationPoint.shortLabel vh.latestBindingTime=postBuild
eventMulticast Subscription Address	ApplicationEndpoint	0..1	ref	Multicast Address that is used by the client to subscribe to the server: This enables the multicast subscription feature. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=eventMulticastSubscription Address.applicationEndpoint, eventMulticastSubscription Address.variationPoint.shortLabel vh.latestBindingTime=postBuild
instance Identifier	AnyServiceInstanceId	0..1	attr	This attribute represents the ability to describe the required service instance ID.
localUnicast Address	ApplicationEndpoint	0..2	ref	The local address over which the CSI is consumed (udp, tcp or both). **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=localUnicastAddress.applicationEndpoint, localUnicastAddress.variationPoint.shortLabel vh.latestBindingTime=postBuild
minorVersion	AnyVersionString	0..1	attr	Minor Version of the ServiceInterface. Value can be set to a number that represents the Minor Version of the searched service or to ANY.

▽

△

Class	**ConsumedServiceInstance**			
providedService Instance	ProvidedService Instance	0..1	ref	Reference to a providedServiceInstance to get the instanceIdentifier information from the ProvidedService Instance. **Tags:**atp.Status=obsolete
remoteUnicast Address	ApplicationEndpoint	0..2	ref	This reference defines the remote address where the service provider is located. This reference shall ONLY be used if the remote address is determined from the configuration and not at runtime from the Service Discovery. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=remoteUnicastAddress.applicationEndpoint, remoteUnicastAddress.variationPoint.shortLabel vh.latestBindingTime=postBuild
sdClientConfig	SdClientConfig	0..1	aggr	Service Discovery Client configuration. **Tags:**atp.Status=obsolete
sdClientTimer Config	SomeipSdClientService InstanceConfig	0..1	ref	Client specific configuration settings relevant for the SOME/IP service discovery. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=sdClientTimerConfig.someipSdClientService InstanceConfig, sdClientTimerConfig.variationPoint.short Label vh.latestBindingTime=postBuild
serviceIdentifier	PositiveInteger	0..1	attr	This attribute represents the ability to describe the SOME/ IP service ID that is searched.
versionDriven FindBehavior	ServiceVersion AcceptanceKindEnum	0..1	attr	Defines the service discovery find behavior. **Tags:**atp.Status=draft

Table A.213: ConsumedServiceInstance

Enumeration	**ContainedIPduCollectionSemanticsEnum**
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication
Note	Defines the collection semantics for ContainedIPdus.
Aggregated by	ContainedIPduProps.collectionSemantics
Literal	*Description*
lastIsBest	The ContainedIPdu data will be fetched via TriggerTransmit just before the transmission executes. **Tags:**atp.EnumerationLiteralIndex=0
queued	The ContainedIPdu data will instantly be stored to the ContainerIPdu in the context of the Transmit API. **Tags:**atp.EnumerationLiteralIndex=1

Table A.214: ContainedIPduCollectionSemanticsEnum

Class	**ContainedIPduProps**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Defines the aspects of an IPdu which can be collected inside a ContainerIPdu.			
Base	*ARObject*			
Aggregated by	ContainerIPdu.containedIPduTriggeringProps, *IPdu*.containedIPduProps			
Attribute	*Type*	*Mult.*	*Kind*	*Note*

▽

△

Class	ContainedIPduProps			
collection Semantics	ContainedIPdu CollectionSemantics Enum	1	attr	Defines whether this ContainedIPdu shall be collected using a last-is-best or queued semantics.
containedPdu Triggering	PduTriggering	0..1	ref	Reference to Pdu for which the ContainedIPduProps are valid.
headerIdLong Header	PositiveInteger	0..1	attr	Defines the header id this IPdu shall have in case this IPdu is put inside a ContainerIPdu with headerType = longHeader.
headerIdShort Header	PositiveInteger	0..1	attr	Defines the header id this IPdu shall have in case this IPdu is put inside a ContainerIPdu with headerType = shortHeader.
offset	PositiveInteger	0..1	attr	Byte offset that describes the location of the Contained Pdu in the ContainerPdu if no header is used.
priority	PositiveInteger	0..1	attr	Defines a priority of a ContainedTxPdu. 255 represents the lowest priority and 0 represent the highest priority.
timeout	TimeValue	0..1	attr	Defines a IPdu specific sender timeout which can reduce the ContainerIPdu timer when this containedIPdu is put inside the ContainerIPdu. This attribute is ignored on receiver side.
trigger	PduCollectionTrigger Enum	0..1	attr	Defines whether this IPdu does trigger the sending of the ContainerIPdu. This attribute is ignored on receiver side.
update IndicationBit Position	PositiveInteger	0..1	attr	The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated.

Table A.215: ContainedIPduProps

Class	ContainerIPdu			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Allows to collect several IPdus in one ContainerIPdu based on the headerType.			
	Tags:atp.recommendedPackage=Pdus			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *IPdu*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Pdu*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
containedIPdu TriggeringProps	ContainedIPduProps	*	aggr	Defines properties for an IPdu that is part of the ContainerIPdu.
containedPdu Triggering	PduTriggering	*	ref	This PduTriggering shall be collected inside the Container IPdu.
container Timeout	TimeValue	0..1	attr	When this timeout expires the ContainerIPdu is sent out. The respective timer is started when the first Ipdu is put into the ContainerIPdu. This attribute is ignored on receiver side.
containerTrigger	ContainerIPduTrigger Enum	0..1	attr	Defines if the transmission of the ContainerIPdu shall be requested right after the first ContainedIPdu was put into it. This attribute shall be ignored on receiver side.
headerType	ContainerIPduHeader TypeEnum	1	attr	Defines whether and which header type is used (header id and length).
minimumRx Container QueueSize	PositiveInteger	0..1	attr	This attribute defines the minimum queue size for received containers.

▽

△

Class	ContainerIPdu			
minimumTx Container QueueSize	PositiveInteger	0..1	attr	This attribute defines the minimum queue size for transmitted containers.
rxAccept ContainedIPdu	RxAcceptContainedI PduEnum	1	attr	Defines whether this ContainerIPdu has a fixed set of containedIPdus assigned for reception.
thresholdSize	PositiveInteger	0..1	attr	Defines the size threshold which, when exceeded, triggers the sending of the ContainerIPdu although the maximum Pdu size has not been reached yet. Unit: byte.
unusedBit Pattern	PositiveInteger	0..1	attr	IPduM fills not updated areas of the ContainerPdu with this byte-pattern.

Table A.216: ContainerIPdu

Enumeration	ContainerIPduHeaderTypeEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication
Note	Is used to define the header type and size of ContainerIPdus. The header size includes the header id and the length information.
Aggregated by	ContainerIPdu.headerType
Literal	Description
longHeader	Header size is 64 bit: • Header Id 32 bit • Dlc 32 bit **Tags:**atp.EnumerationLiteralIndex=0
noHeader	No Header is used and the location of each containedPdu in the ContainerPdu is statically configured. **Tags:**atp.EnumerationLiteralIndex=2
shortHeader	Header size is 32 bit: • Header Id 24 bit • Dlc 8 bit. **Tags:**atp.EnumerationLiteralIndex=1

Table A.217: ContainerIPduHeaderTypeEnum

Class	CouplingElement			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	A CouplingElement is used to connect EcuInstances to the VLAN of an EthernetCluster. Coupling Elements can reach from a simple hub to a complex managed switch or even devices with functionalities in higher layers. A CouplingElement that is not related to an EcuInstance occurs as a dedicated single device. **Tags:**atp.recommendedPackage=CouplingElements			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
communication Cluster	EthernetCluster	1	ref	This relationship defines to which cluster the Coupling Element belongs.

▽

$\triangle$

Class	CouplingElement			
couplingPort	CouplingPort	*	aggr	Hardware Port of the CouplingElement that is used to connect this CouplingPort to EcuInstances or other CouplingElements. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=couplingPort.shortName, coupling Port.variationPoint.shortLabel vh.latestBindingTime=postBuild
couplingType	CouplingElementEnum	1	attr	Describes the coupling type of this CouplingElement.
ecuInstance	EcuInstance	0..1	ref	Optional reference to the ECU where the Coupling Element is located.

Table A.218: CouplingElement

Enumeration	CouplingElementEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology
Note	Identifies the Coupling type.
Aggregated by	CouplingElement.couplingType
Literal	Description
hub	A device that is used to connect segments of a LAN. In Hubs frames are "broadcasted" to every one of its ports. **Tags:**atp.EnumerationLiteralIndex=0
router	A device that routes frames between different networks. **Tags:**atp.EnumerationLiteralIndex=1
switch	A device that filters and forwards frames between different LAN segments. **Tags:**atp.EnumerationLiteralIndex=2

Table A.219: CouplingElementEnum

Class	CouplingPort			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	A CouplingPort is used to connect a CouplingElement with an EcuInstance or two CouplingElements with each other via a CouplingPortConnection. Optionally, the CouplingPort may also have a reference to a macMulticastGroup and a defaultVLAN.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	CouplingElement.couplingPort, EthernetCommunicationController.couplingPort			
Attribute	Type	Mult.	Kind	Note
connection Negotiation Behavior	EthernetConnection NegotiationEnum	0..1	attr	Specifies the connection negotiation of the CouplingPort.
couplingPort Details	CouplingPortDetails	0..1	aggr	Defines more details of a CouplingPort in case a more specific configuration is required.
couplingPort Role	CouplingPortRoleEnum	0..1	attr	Defines the role this CouplingPort takes in the context of the CouplingElement.

$\triangledown$

△

Class	CouplingPort			
defaultVlan	EthernetPhysical Channel	0..1	ref	The vLanIdentifier of the referenced VLAN is the Default-PVID (port VLAN ID). A Port VLAN ID is a default VLAN ID that is assigned to an access CouplingPort to designate the VLAN segment to which this port is connected. Also, if a CouplingPort has not been configured with any VLAN memberships, the virtual switch's Port VLAN ID (pvid) becomes the default VLAN ID for the ports connection. This identifier/tag is added for incoming untagged messages at the port (ingress tagging). For outgoing messages with this identifier, the tag is removed at the port (egress untagging, depending on the Vlan Membership.sendActivity).
macLayerType	EthernetMacLayerType Enum	0..1	attr	Specifies the mac layer type of the CouplingPort.
macMulticast Address	MacMulticastGroup	*	ref	Assigns a set of MAC-Multicast-Addresses which are addressable via this CouplingPort. This is a static pre-configuration and further addresses may be learned during runtime.
macSecProps	MacSecProps	*	aggr	Properties to configure MACsec (Media access control security) and the MKA (MACsec Key Agreement) for the CouplingPort (PHY). **Tags:**atp.Status=candidate
physicalLayer Type	EthernetPhysicalLayer TypeEnum	0..1	attr	Specifies the physical layer type of the CouplingPort.
plcaProps	PlcaProps	0..1	aggr	Optional properties for configuration of PLCA (Physical Layer Collision Avoidance) in case 10-BASE-T1S Ethernet is used and PLCA is enabled on the Coupling Port (PHY).
pncMapping	PncMappingIdent	*	ref	Reference to the partial networks this CouplingPort participates in.
receiveActivity	EthernetSwitchVlan IngressTagEnum	0..1	attr	Defines the handling of frames at the ingress port.
vlan Membership	VlanMembership	*	aggr	Messages of VLANs that are defined here can be communicated via the CouplingPort.
vlanModifier	EthernetPhysical Channel	0..1	ref	All incoming messages at this CouplingPort shall be tagged with this VLAN Id. This tagging is performed regardless whether the message already has a VLAN tag or is untagged, an existing VLAN tag will be overwritten. This feature is XOR with CoupligPort.defaultVlan.
wakeupSleep OnDataline Config	EthernetWakeupSleep OnDatalineConfig	0..1	ref	Optional reference to EthernetWakeupSleepOnDataline Config.

Table A.220: CouplingPort

Class	CouplingPortConnection			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Connection between two CouplingPorts (firstPort and secondPort) or between a collection of Ports that are all referenced by the portCollection reference.			
Base	ARObject			
Aggregated by	EthernetCluster.couplingPortConnection			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	**CouplingPortConnection**			
firstPort	CouplingPort	0..1	ref	Reference to the first CouplingPort that is connected via the CouplingPortConnection.
nodePort	CouplingPort	*	ref	Reference to a number of CouplingPorts that are connected via the CouplingPortConnection. This reference shall be used to describe a 10BASE-T1S topology architecture where several CouplingPorts of EthernetCommunicationControllers are connected via one CouplingPortConnection. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=nodePort.couplingPort, nodePort.variation Point.shortLabel vh.latestBindingTime=postBuild
plcaLocalNode Count	PositiveInteger	0..1	attr	Defines the number of communication participants in case 10BASE-T1S and the nodePort reference is used.
plcaTransmit Opportunity Timer	PositiveInteger	0..1	attr	Timer for the transmission in bit time to evaluate if a Transmission Opportunity is yield or not.
secondPort	CouplingPort	0..1	ref	Reference to the second CouplingPort that is connected via the CouplingPortConnection.

Table A.221: CouplingPortConnection

Class	**CouplingPortDetails**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Defines details of a CouplingPort. May be used to configure the structures of a switch.			
Base	ARObject			
Aggregated by	CouplingPort.couplingPortDetails			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
couplingPort Structural Element	CouplingPortStructural Element	1..*	aggr	Collects all the structural parts at which a CouplingPort may be configurable.
ethernetPriority Regeneration	EthernetPriority Regeneration	0..8	aggr	Defines a priority regeneration where the ingress priority is replaced by regenerated priority.
ethernetTraffic Class Assignment	CouplingPortTraffic ClassAssignment	0..8	aggr	Defines the ingress port to EthernetTrafficClass assignment.
globalTime Props	GlobalTimeCoupling PortProps	0..1	aggr	Specifies properties for the usage of the CouplingPort in the scope of Global Time Sync.
lastEgress Scheduler	CouplingPortScheduler	0..1	ref	Defines which CouplingPortScheduler is the last in the egress port structure.
ratePolicy	CouplingPortRatePolicy	*	aggr	Rate policies to be applied for this CouplingPort.

Table A.222: CouplingPortDetails

Enumeration	**CouplingPortRoleEnum**
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology
Note	Defines the role a CouplingPort takes in the context of a CouplingElement.
Aggregated by	CouplingPort.couplingPortRole

▽

△

Enumeration	CouplingPortRoleEnum
Literal	Description
hostPort	The hostPort is connected to an ECU (host ecu). The host ECU controls the connected Coupling Element (e.g. Ethernet switch). **Tags:**atp.EnumerationLiteralIndex=0
standardPort	A CoupingPort can be a standardPort that is used to connect the CouplingElement with Coupling Ports outside the ECU. **Tags:**atp.EnumerationLiteralIndex=2
upLinkPort	A CouplingPort can be connected to another CouplingPort of a CouplingElement located on the same ECU (CouplingElement.ecuInstance) using the CouplingPortConnection. This is used to model a cascaded switch. **Tags:**atp.EnumerationLiteralIndex=1

Table A.223: CouplingPortRoleEnum

Class	CouplingPortTrafficClassAssignment			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Defines the assignment of Traffic Class to a frame.			
Base	ARObject, *Referrable*			
Aggregated by	CouplingPortDetails.ethernetTrafficClassAssignment			
Attribute	Type	Mult.	Kind	Note
priority	PositiveInteger	0..8	attr	Defines a priority which is mapped onto a Traffic Class.
trafficClass	PositiveInteger	1	attr	Defines the Traffic Class which is assigned. range: 0-7

Table A.224: CouplingPortTrafficClassAssignment

Class	CpSoftwareCluster			
Package	M2::AUTOSARTemplates::SystemTemplate::SoftwareCluster			
Note	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. **Tags:** atp.Status=draft atp.recommendedPackage=CpSoftwareClusters			
Base	*ARElement*, ARObject, CollectableElement, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
softwareCluster Id	PositiveInteger	0..1	attr	This attribute represents the value of the id of the corresponding CP software cluster.
swComponent Assignment	SwComponent PrototypeAssignment	*	aggr	This is the collection of SwComponentPrototype Assignments **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=swComponentAssignment, swComponent Assignment.variationPoint.shortLabel atp.Status=draft vh.latestBindingTime=postBuild

▽

△

Class	CpSoftwareCluster			
swComposition	CompositionSw ComponentType	*	ref	Software Components in the context of a CompositionSw ComponentType 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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=swComposition.compositionSwComponent Type, swComposition.variationPoint.shortLabel atp.Status=draft vh.latestBindingTime=systemDesignTime

Table A.225: CpSoftwareCluster

Class	CpSoftwareClusterBinaryManifestDescriptor			
Package	M2::AUTOSARTemplates::SystemTemplate::SoftwareCluster::BinaryManifest			
Note	This meta-class has the ability to act as a hub for all information related to the binary manifest of a given CP software cluster. The manifest is subject to integrator work and therefore not a part of the definition of the CP software cluster itself. **Tags:** atp.Status=draft atp.recommendedPackage=CpSoftwareClusterBinaryManifestDescriptors			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
cpSoftware Cluster	CpSoftwareCluster	0..1	ref	This reference identifies the CpSoftwareCluster to which the enclosing CpSoftwareClusterBinaryManifest Descriptor belongs, The CpSoftwareClusterBinaryManifestDescriptor is defined in an integration phase while the referenced Cp SoftwareCluster represents a design element. Therefore, it makes sense to use a reference rather than an aggregation in the relation of the two meta-classes. **Tags:**atp.Status=draft
metaDataField	BinaryManifestMeta DataField	*	aggr	This aggregation identifies the collection of meta-data contained in the enclosing binary manifest. **Tags:**atp.Status=draft
provide Resource	BinaryManifestProvide Resource	*	aggr	This aggregation represents the collection of provided resources in the enclosing binary manifest. **Tags:**atp.Status=draft
require Resource	BinaryManifestRequire Resource	*	aggr	This aggregation represents the collection of required resources in the enclosing binary manifest. **Tags:**atp.Status=draft
resource Definition	BinaryManifest ResourceDefinition	*	aggr	This aggregation represents the collection of binary manifest resource definitions that belong to the enclosing CpSoftwareClusterBinaryManifestDescriptor. **Tags:**atp.Status=draft

▽

△

Class	CpSoftwareClusterBinaryManifestDescriptor			
softwareCluster Id	PositiveInteger	0..1	attr	This attribute represents the value of the id of the corresponding CP software cluster. This id is assigned by an integrator, but may also be copied from CpSoftware Cluster.softwareClusterId if available. **Tags:**atp.Status=draft

Table A.226: CpSoftwareClusterBinaryManifestDescriptor

Class	CpSoftwareClusterCommunicationResource			
Package	M2::AUTOSARTemplates::SystemTemplate::SoftwareCluster			
Note	Represents a single resource required or provided by a CP Software Cluster which relates to the port based communication on VFB level. **Tags:**atp.Status=draft			
Base	*ARObject*, *CpSoftwareClusterResource*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CpSoftwareClusterResourcePool.resource			
Attribute	Type	Mult.	Kind	Note
communication ResourceProps	CpSoftwareCluster Communication ResourceProps	0..1	aggr	This aggregation supports the further qualification of the enclosing CpSoftwareClusterCommunicationRecource by means of additional attributes depending on the nature of the CpSoftwareClusterCommunicationRecource.

Table A.227: CpSoftwareClusterCommunicationResource

Class	*CpSoftwareClusterResource* (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::SoftwareCluster			
Note	Represents a single resource required or provided by a CP Software Cluster. **Tags:** atp.Status=draft atp.recommendedPackage=Resources			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	CpSoftwareClusterCommunicationResource, CpSoftwareClusterServiceResource			
Aggregated by	CpSoftwareClusterResourcePool.resource			
Attribute	Type	Mult.	Kind	Note
dependent Resource	RoleBasedResource Dependency	*	aggr	Link to a resource which depends on this resource to implement them. **Tags:**atp.Status=draft
globalResource Id	PositiveInteger	0..1	attr	A unique identifiers per resource used for the connection process. The identifier is required to be unique in the scope of a single machine. If software clusters are designed to be reused on multiple machines the uniqueness requirements applies for all the intended machines. **Tags:**atp.Status=draft
isMandatory	Boolean	0..1	attr	This attribute indicates, that the resource is mandatory to operate the Software Cluster. If the resource is not provided on the machine the connection process of any Software Cluster requiring this resource gets aborted. **Tags:**atp.Status=draft

Table A.228: CpSoftwareClusterResource

Class	CpSoftwareClusterResourceToApplicationPartitionMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::SoftwareCluster			
Note	This meta class maps a Software Cluster resource to an Application Partition to restrict the usage. **Tags:**atp.Status=draft			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CpSoftwareClusterMappingSet.resourceToApplicationPartitionMapping, SystemMapping.resourceTo ApplicationPartitionMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
application Partition	ApplicationPartition	0..1	ref	ApplicationPartition for which the mapping applies. **Tags:**atp.Status=draft
resource	CpSoftwareCluster Resource	0..1	ref	Software Cluster Resource for which the mapping applies. **Tags:**atp.Status=draft

Table A.229: CpSoftwareClusterResourceToApplicationPartitionMapping

Class	CpSoftwareClusterServiceResource			
Package	M2::AUTOSARTemplates::SystemTemplate::SoftwareCluster			
Note	Represents a single resource required or provided by a CP Software Cluster which relates to the BSW. **Tags:**atp.Status=draft			
Base	*ARObject*, *CpSoftwareClusterResource*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	CpSoftwareClusterResourcePool.resource			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
resourceNeeds	EcucContainerValue	*	ref	Refernce(s) to one or multiple EcucContainerValue(s) qualifying the characteristics of the resource. **Tags:**atp.Status=draft

Table A.230: CpSoftwareClusterServiceResource

Class	CpSoftwareClusterToEcuInstanceMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::SoftwareCluster			
Note	This meta class maps a CpSoftwareCluster to a EcuInstance. **Tags:**atp.Status=draft			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SystemMapping.swClusterMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
ecuInstance	EcuInstance	0..1	ref	Reference to a specific ECU Instance description. **Tags:**atp.Status=draft
machineId	PositiveInteger	0..1	attr	Unique number of the (virtual or physical) machine to which the Software Cluster is mapped.
swCluster	CpSoftwareCluster	*	ref	The mapped CP Software Cluster **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=swCluster.cpSoftwareCluster, sw Cluster.variationPoint.shortLabel atp.Status=draft vh.latestBindingTime=systemDesignTime

Table A.231: CpSoftwareClusterToEcuInstanceMapping

Class	CpSoftwareClusterToResourceMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::SoftwareCluster			
Note	This meta class maps a service resource to CP Software Clusters. By this mapping it's specified whether the Software Cluster has to provide or to require the resource. **Tags:**atp.Status=draft			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	CpSoftwareClusterMappingSet.softwareClusterToResourceMapping, SystemMapping.softwareClusterToResourceMapping			
Attribute	Type	Mult.	Kind	Note
provider	CpSoftwareCluster	0..1	ref	CP Software Cluster providing the resource **Tags:**atp.Status=draft
requester	CpSoftwareCluster	*	ref	CP Software Cluster requesting the resource **Tags:**atp.Status=draft
service Resource	CpSoftwareCluster ServiceResource	0..1	ref	Service resource for which the mapping applies. **Tags:**atp.Status=draft

Table A.232: CpSoftwareClusterToResourceMapping

Class	CryptoServiceCertificate			
Package	M2::AUTOSARTemplates::SystemTemplate::SecureCommunication			
Note	This meta-class represents the ability to model a cryptographic certificate. **Tags:**atp.recommendedPackage=CryptoServiceCertificates			
Base	ARElement, ARObject, CollectableElement, Identifiable, MultilanguageReferrable, Packageable Element, Referrable			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
algorithmFamily	CryptoCertificate AlgorithmFamilyEnum	0..1	attr	This attribute represents a description of the family of crypto algorithm used to generate public key and signature of the cryptographic certificate.
format	CryptoCertificateFormat Enum	0..1	attr	This attribute can be used to provide information about the format used to create the certificate
maximum Length	PositiveInteger	0..1	attr	This attribute represents the ability to define the maximum length of the certificate in bytes.
nextHigher Certificate	CryptoService Certificate	0..1	ref	The reference identifies the next higher certificate in the certificate chain.
serverName Identification	String	0..1	attr	Server Name Indication (SNI) is needed if the IP address hosts multiple servers (on the same port), each of them using a different certificate. If the client sends the SNI to the Server in the client hello, the server looks the SNI up in its certificate list and uses the certificate identified by the SNI.

Table A.233: CryptoServiceCertificate

Class	CryptoServiceKey	
Package	M2::AUTOSARTemplates::SystemTemplate::SecureCommunication	
Note	This meta-class has the ability to represent a crypto key. **Tags:**atp.recommendedPackage=CryptoDevelopmentKeys	
Base	ARElement, ARObject, CollectableElement, Identifiable, MultilanguageReferrable, Packageable Element, Referrable	

▽

△

Class	CryptoServiceKey			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
algorithmFamily	String	1	attr	This attribute represent the description of the family of the applicable crypto algorithm.
development Value	ValueSpecification	0..1	aggr	This aggregation represents the ability to assign a specific value to the crypto key as part of the system description. This value can then be taken for the development of the respective ECU.
keyGeneration	CryptoServiceKey GenerationEnum	0..1	attr	This attribute describes how a the specific cryptographic key is created.
keyStorageType	String	0..1	attr	This attribute describes where the enclosing cryptographic key shall be stored. AUTOSAR reserves specific values for this attributes but it is possible to insert custom values as well.
length	PositiveInteger	1	attr	This attribute describes the length of the cryptographic key in bits.

Table A.234: CryptoServiceKey

Class	CryptoServicePrimitive			
Package	M2::AUTOSARTemplates::SystemTemplate::SecureCommunication			
Note	This meta-class has the ability to represent a crypto primitive. **Tags:**atp.recommendedPackage=CryptoPrimitives			
Base	*ARElement*, ARObject, CollectableElement, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
algorithmFamily	String	0..1	attr	This attribute represents a description of the family (e.g. AES) of crypto algorithm implemented by the crypto primitive.
algorithmMode	String	0..1	attr	This attribute represents a description of the mode of the crypto algorithm implemented by the crypto primitive.
algorithm Secondary Family	String	0..1	attr	This attribute represents a further description of the secondary family of crypto algorithm implemented by the crypto primitive. The secondary family is needed for the specification of the hash algorithm for a signature check, e.g. using RSA.

Table A.235: CryptoServicePrimitive

Class	CryptoServiceQueue			
Package	M2::AUTOSARTemplates::SystemTemplate::SecureCommunication			
Note	This meta-class has the ability to represent a crypto queue. **Tags:**atp.recommendedPackage=CryptoServiceQueues			
Base	*ARElement*, ARObject, CollectableElement, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
queueSize	PositiveInteger	0..1	attr	Defines the queue size of the CryptoServiceQueue.

Table A.236: CryptoServiceQueue

Class	DataConstr			
Package	M2::MSR::AsamHdo::Constraints::GlobalConstraints			
Note	This meta-class represents the ability to specify constraints on data. **Tags:**atp.recommendedPackage=DataConstrs			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpBlueprintable*, *CollectableElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
dataConstrRule	DataConstrRule	*	aggr	This is one particular rule within the data constraints. **Tags:** xml.roleElement=true xml.roleWrapperElement=true xml.sequenceOffset=30 xml.typeElement=false xml.typeWrapperElement=false

Table A.237: DataConstr

Class	DataConstrRule			
Package	M2::MSR::AsamHdo::Constraints::GlobalConstraints			
Note	This meta-class represents the ability to express one specific data constraint rule.			
Base	*ARObject*			
Aggregated by	DataConstr.dataConstrRule			
Attribute	Type	Mult.	Kind	Note
constrLevel	Integer	0..1	attr	This attribute describes the category of a constraint. One of its functions is in the area of constraint violation, where it can be used from a certain level, to produce error messages. The lower the level, the more stringent the check. Used to distinguish hard or soft limits. **Tags:**xml.sequenceOffset=20
internalConstrs	InternalConstrs	0..1	aggr	Describes the limitations applicable on the internal domain (as opposed to the physical domain). **Tags:**xml.sequenceOffset=40
physConstrs	PhysConstrs	0..1	aggr	Describes the limitations applicable on the physical domain (as opposed to the internal domain). **Tags:**xml.sequenceOffset=30

Table A.238: DataConstrRule

Class	DataFilter			
Package	M2::AUTOSARTemplates::CommonStructure::Filter			
Note	Base class for data filters. The type of the filter is specified in attribute dataFilterType. Some of the filter types require additional arguments which are specified as attributes of this class.			
Base	*ARObject*			
Aggregated by	ISignalPort.dataFilter, NonqueuedReceiverComSpec.filter, NonqueuedSenderComSpec.dataFilter, SignalBasedEventElementToISignalTriggeringMapping.filter, SignalBasedFieldToISignalTriggering Mapping.filter, SignalServiceTranslationElementProps.filter, TransmissionModeCondition.dataFilter			
Attribute	Type	Mult.	Kind	Note
dataFilterType	DataFilterTypeEnum	0..1	attr	This attribute specifies the type of the filter.
mask	UnlimitedInteger	0..1	attr	Mask for old and new value.

▽

$\triangle$

Class	DataFilter			
max	UnlimitedInteger	0..1	attr	Value to specify the upper boundary
min	UnlimitedInteger	0..1	attr	Value to specify the lower boundary
offset	PositiveInteger	0..1	attr	Specifies the initial number of messages to occur before the first message is passed
period	PositiveInteger	0..1	attr	Specifies number of messages to occur before the message is passed again
x	UnlimitedInteger	0..1	attr	Value to compare with

Table A.239: DataFilter

Enumeration	DataFilterTypeEnum
Package	M2::AUTOSARTemplates::CommonStructure::Filter
Note	This enum specifies the supported DataFilterTypes.
Aggregated by	DataFilter.dataFilterType
Literal	Description
always	No filtering is performed so that the message always passes.
	Tags:atp.EnumerationLiteralIndex=0
maskedNewDiffers MaskedOld	Pass messages where the masked value has changed.
	(new_value&mask) !=(old_value&mask)
	new_value: current value of the message
	old_value: last value of the message (initialized with the initial value of the message, updated with new_value if the new message value is not filtered out)
	Tags:atp.EnumerationLiteralIndex=1
maskedNewDiffers X	Pass messages whose masked value is not equal to a specific value x
	(new_value&mask) != x
	new_value: current value of the message
	Tags:atp.EnumerationLiteralIndex=2
maskedNewEquals X	Pass messages whose masked value is equal to a specific value x
	(new_value&mask) == x
	new_value: current value of the message
	Tags:atp.EnumerationLiteralIndex=3
never	The filter removes all messages.
	Tags:atp.EnumerationLiteralIndex=4
newIsOutside	Pass a message if its value is outside a predefined boundary.
	(min > new_value) OR (new_value > max)
	Tags:atp.EnumerationLiteralIndex=5
newIsWithin	Pass a message if its value is within a predefined boundary.
	min <= new_value <= max
	Tags:atp.EnumerationLiteralIndex=6

$\triangledown$

△

Enumeration	DataFilterTypeEnum
oneEveryN	Pass a message once every N message occurrences. Algorithm: occurrence % period == offset Start: occurrence = 0. Each time the message is received or transmitted, occurrence is incremented by 1 after filtering. Length of occurrence is 8 bit (minimum). **Tags:**atp.EnumerationLiteralIndex=7

Table A.240: DataFilterTypeEnum

Enumeration	DataIdModeEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer
Note	Supported inclusion modes to include the implicit two-byte Data ID in the one-byte CRC.
Aggregated by	E2EProfileConfiguration.dataIdMode, EndToEndTransformationDescription.dataIdMode
Literal	*Description*
all16Bit	Two bytes are included in the CRC (double ID configuration). **Tags:**atp.EnumerationLiteralIndex=0
alternating8Bit	One of the two bytes byte is included, alternating high and low byte, depending on parity of the counter (alternating ID configuration). For even counter low byte is included; For odd counters the high byte is included. **Tags:**atp.EnumerationLiteralIndex=1
lower12Bit	The low byte is included in the implicit CRC calculation, the low nibble of the high byte is transmitted along with the data (i.e. it is explicitly included), the high nibble of the high byte is not used. This is applicable for the IDs up to 12 bits. **Tags:**atp.EnumerationLiteralIndex=2
lower8Bit	Only low byte is included, high byte is never used. This is applicable if the IDs in a particular system are 8 bits. **Tags:**atp.EnumerationLiteralIndex=3

Table A.241: DataIdModeEnum

Class	DataInterface (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	The purpose of this meta-class is to act as an abstract base class for subclasses that share the semantics of being concerned about data (as opposed to e.g. operations).			
Base	*ARElement*, ARObject, *AtpBlueprint*, AtpBlueprintable, *AtpClassifier*, *AtpType*, CollectableElement, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *PortInterface*, *Referrable*			
Subclasses	NvDataInterface, ParameterInterface, SenderReceiverInterface			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
–	–	–	–	–

Table A.242: DataInterface

Class	DataMapping (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	Mapping of port elements (data elements and parameters) to frames and signals.			
Base	ARObject			
Subclasses	ClientServerToSignalMapping, SenderReceiverCompositeElementToSignalMapping, SenderReceiverToSignalGroupMapping, SenderReceiverToSignalMapping, TriggerToSignalMapping			
Aggregated by	SystemMapping.dataMapping			
Attribute	Type	Mult.	Kind	Note
introduction	DocumentationBlock	0..1	aggr	This represents introductory documentation about the data mapping.

Table A.243: DataMapping

Class	DataPrototype (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::DataPrototypes			
Note	Base class for prototypical roles of any data type.			
Base	ARObject, AtpFeature, AtpPrototype, Identifiable, MultilanguageReferrable, Referrable			
Subclasses	ApplicationCompositeElementDataPrototype, AutosarDataPrototype			
Aggregated by	AtpClassifier.atpFeature			
Attribute	Type	Mult.	Kind	Note
swDataDefProps	SwDataDefProps	0..1	aggr	This property allows to specify data definition properties which apply on data prototype level. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=swDataDefProps

Table A.244: DataPrototype

Class	DataPrototypeInSenderReceiverInterfaceInstanceRef			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer::InstanceRef			
Note				
Base	ARObject, AtpInstanceRef, DataPrototypeInPortInterfaceInstanceRef			
Aggregated by	DataPrototypeInPortInterfaceRef.dataPrototypeInSenderReceiverInterface			
Attribute	Type	Mult.	Kind	Note
base	SenderReceiverInterface	0..1	ref	**Stereotypes:** atpDerived
contextDataPrototypeInSr (ordered)	ApplicationCompositeElementDataPrototype	*	ref	**Tags:**xml.sequenceOffset=20
rootDataPrototypeInSr	AutosarDataPrototype	0..1	ref	**Tags:**xml.sequenceOffset=10
targetDataPrototypeInSr	DataPrototype	1	ref	**Tags:**xml.sequenceOffset=30

Table A.245: DataPrototypeInSenderReceiverInterfaceInstanceRef

Class	DataPrototypeMapping				
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface				
Note	Defines the mapping of two particular VariableDataPrototypes, ParameterDataPrototypes or Argument DataPrototypes with non-equal shortNames, non-equal structure (specific condition is described by [constr_1187]), and/or non-equal semantic (resolution or range) in context of two different Sender ReceiverInterface, NvDataInterface or ParameterInterface or Operations.				
	If the semantic is unequal, the following rules apply: The textTableMapping is only applicable if the referred DataPrototypes are typed by AutosarDataType referring to CompuMethods of category TEXTTABLE, SCALE_LINEAR_AND_TEXTTABLE or BITFIELD_TEXTTABLE.				
	In the case that the DataPrototypes are typed by AutosarDataType either referring to CompuMethods of category LINEAR, IDENTICAL or referring to no CompuMethod (which is similar as IDENTICAL) the linear conversion factor is calculated out of the factorSiToUnit and offsetSiToUnit attributes of the referred Units and the CompuRationalCoeffs of a compuInternalToPhys of the referred CompuMethods.				
Base	ARObject				
Aggregated by	ClientServerOperationMapping.argumentMapping, VariableAndParameterInterfaceMapping.dataMapping				
Attribute	Type	Mult.	Kind	Note	
firstData Prototype	AutosarDataPrototype	0..1	ref	First to be mapped DataPrototype in context of a Sender ReceiverInterface, NvDataInterface, ParameterInterface or Operation.	
firstToSecond Data Transformation	DataTransformation	0..1	ref	This reference defines the need to execute the Data Transformation <Mip>_<transformerId> functions of the transformation chain when communicating from the Data PrototypeMapping.firstDataPrototype to the Data PrototypeMapping.secondDataPrototype.	
				This reference also specifies the reverse Data Transformation <Mip>_Inv_<transformerId> functions of the transformation chain (i.e. from the DataPrototype Mapping.secondDataPrototype to the DataPrototype Mapping.firstDataPrototype) if the referenced Data Transformation is symmetric, i.e. attribute Data Transformation.dataTransformationKind is set to symmetric.	
secondData Prototype	AutosarDataPrototype	0..1	ref	Second to be mapped DataPrototype in context of a SenderReceiverInterface, NvDataInterface, Parameter Interface or Operation.	
secondToFirst Data Transformation	DataTransformation	0..1	ref	This defines the need to execute the reverse Data Transformation <Mip>_Inv_<transformerId> functions of the transformation chain when communicating from the DataPrototypeMapping.secondDataPrototype to the Data PrototypeMapping.firstDataPrototype.	
subElement Mapping	SubElementMapping	*	aggr	This represents the owned SubelementMapping.	
textTable Mapping	TextTableMapping	0..2	aggr	Applied TextTableMapping(s)	

Table A.246: DataPrototypeMapping

Class	DataPrototypeReference (abstract)				
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer				
Note	This meta-class provides the ability to reference a DataPrototype.				
Base	ARObject				
Subclasses	DataPrototypeInPortInterfaceRef, ImplementationDataTypeElementInPortInterfaceRef				
Aggregated by	DataPrototypeTransformationProps.dataPrototypeInPortInterfaceRef, SignalServiceTranslationElement Props.element				
Attribute	Type	Mult.	Kind	Note	

▽

△

Class	DataPrototypeReference (abstract)			
tagId	PositiveInteger	0..1	attr	This attribute represents the ability to specify a tag-id for the serialization of a specific DataPrototype in the context of a (potentially deeply-nested) composite data structure.

Table A.247: DataPrototypeReference

Class	DataPrototypeTransformationProps			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	DataPrototypeTransformationProps allows to set the attributes for the different Transformation Technologies that are DataPrototype specific.			
Base	ARObject			
Aggregated by	TransformationISignalProps.dataPrototypeTransformationProps			
Attribute	Type	Mult.	Kind	Note
dataPrototypeInPortInterfaceRef	DataPrototypeReference	0..1	aggr	Reference to a DataPrototype that is transported in the serialized ISignal.
networkRepresentationProps	SwDataDefProps	0..1	aggr	Specification of the actual network representation for the referenced primitive DataPrototype. If a network representation is provided then the baseType shall be used by the Transformer as input for the serialization/deserilaization. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=networkRepresentationProps
transformationProps	TransformationProps	0..1	ref	Collection of AutosarDataPrototype related configuration settings for a transformer.

Table A.248: DataPrototypeTransformationProps

Class	DataReceiveErrorEvent			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents			
Note	This event is raised when the Com layer detects and notifies an error concerning the reception of the referenced ValiableDataPrototype.			
Base	ARObject, AbstractEvent, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable, MultilanguageReferrable, RTEEvent, Referrable			
Aggregated by	AtpClassifier.atpFeature, SwcInternalBehavior.event			
Attribute	Type	Mult.	Kind	Note
data	VariableDataPrototype	0..1	iref	The referenced VariableDataPrototype raises this Data ReceiveErrorEvent when there was an error during the reception. **InstanceRef implemented by:**RVariableInAtomicSwcInstanceRef

Table A.249: DataReceiveErrorEvent

Class	DataReceivedEvent	
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents	
Note	This event is raised when the referenced data element is received.	
Base	ARObject, AbstractEvent, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable, MultilanguageReferrable, RTEEvent, Referrable	
Aggregated by	AtpClassifier.atpFeature, SwcInternalBehavior.event	

▽

△

Class	DataReceivedEvent			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
data	VariableDataPrototype	0..1	iref	The referenced VariableDataPrototype raises this Data ReceivedEvent when the data has been received.
				InstanceRef implemented by:RVariableInAtomicSwc InstanceRef

Table A.250: DataReceivedEvent

Class	DataSendCompletedEvent			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents			
Note	This event is raised when the referenced explicit data element has been sent or an error occurred.			
Base	*ARObject*, *AbstractEvent*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *Multilanguage Referrable*, *RTEEvent*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, SwcInternalBehavior.event			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
eventSource	VariableAccess	0..1	ref	The referenced VariableAccess raises this DataSend CompletedEvent when the explicit write access was successful or an error occurred.

Table A.251: DataSendCompletedEvent

Class	DataTransformation			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	A DataTransformation represents a transformer chain. It is an ordered list of transformers.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DataTransformationSet.dataTransformation			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
data Transformation Kind	DataTransformationKind Enum	0..1	attr	This attribute controls the kind of DataTransformation to be applied.
executeDespite Data Unavailability	Boolean	1	attr	Specifies whether the transformer chain is executed even if no input data are available.
transformer Chain (ordered)	Transformation Technology	1..*	ref	This attribute represents the definition of a chain of transformers that are supposed to be executed according to the order of being referenced from DataTransformation.

Table A.252: DataTransformation

Enumeration	DataTransformationKindEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer
Note	This enumeration contributes to the definition of the scope of the DataTransformation.
Aggregated by	DataTransformation.dataTransformationKind
Literal	**Description**
asymmetricFrom ByteArray	The DataTransformation shall only be applied to the receiving end only, i.e. transform from byte array to data type. **Tags:**atp.EnumerationLiteralIndex=0

▽

Document ID 635: AUTOSAR_TR_ClassicAutosarModelConstraints

△

Enumeration	DataTransformationKindEnum
asymmetricToByteArray	The DataTransformation shall be applied to the sending end only, i.e. from data type to byte array. **Tags:**atp.EnumerationLiteralIndex=1
symmetric	The DataTransformation shall be applied at both the sending and the receiving end of the communication. **Tags:**atp.EnumerationLiteralIndex=2

Table A.253: DataTransformationKindEnum

Class	DataTypeMap			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::Datatypes			
Note	This class represents the relationship between ApplicationDataType and its implementing Abstract ImplementationDataType.			
Base	ARObject			
Aggregated by	DataTypeMappingSet.dataTypeMap			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
applicationDataType	ApplicationDataType	0..1	ref	This is the corresponding ApplicationDataType
implementationDataType	AbstractImplementationDataType	0..1	ref	This is the corresponding AbstractImplementationDataType.

Table A.254: DataTypeMap

Class	DataTypeMappingSet			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::Datatypes			
Note	This class represents a list of mappings between ApplicationDataTypes and ImplementationDataTypes. In addition, it can contain mappings between ImplementationDataTypes and ModeDeclarationGroups. **Tags:**atp.recommendedPackage=DataTypeMappingSets			
Base	*ARElement*, ARObject, *AtpBlueprint*, AtpBlueprintable, CollectableElement, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dataTypeMap	DataTypeMap	*	aggr	This is one particular association between an Application DataType and its AbstractImplementationDataType.
modeRequestTypeMap	ModeRequestTypeMap	*	aggr	This is one particular association between an Mode DeclarationGroup and its AbstractImplementationDataType.

Table A.255: DataTypeMappingSet

Enumeration	DataTypePolicyEnum
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping
Note	This class lists the supported DataTypePolicies.
Aggregated by	ISignal.dataTypePolicy
Literal	**Description**

▽

△

Enumeration	DataTypePolicyEnum
legacy	In case the System Description doesn't use a complete Software Component Description (VFB View) this value can be chosen. This supports the inclusion of legacy signals. The aggregation of SwDataDefProps shall be used to configure the "ComSignalDataInvalidValue" and the Data Semantics. **Tags:**atp.EnumerationLiteralIndex=0
network Representation FromComSpec	Ignore any networkRepresentationProps of this ISignal and use the networkRepresentation from the ComSpec. Please note that the usage does not imply the existence of the SwDataDefProps in the role network Representation aggregated by the SenderComSpec or ReceiverComSpec if an ImplementationData Type is defined. **Tags:**atp.EnumerationLiteralIndex=1
override	If this value is chosen the requirements specified in the ComSpec (networkRepresentationFromCom Spec) are not fullfilled by the aggregated SwDataDefProps. In this case the networkRepresentation is specified by the aggregated swDataDefProps. **Tags:**atp.EnumerationLiteralIndex=2
transformingISignal	This literal indicates that a transformer chain shall be used to communicate the ISignal as UINT8_N over the bus. **Tags:**atp.EnumerationLiteralIndex=4

Table A.256: DataTypePolicyEnum

Class	DataWriteCompletedEvent			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents			
Note	This event is raised when an implicit write access was successful or an error occurred.			
Base	ARObject, AbstractEvent, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable, Multilanguage Referrable, RTEEvent, Referrable			
Aggregated by	AtpClassifier.atpFeature, SwcInternalBehavior.event			
Attribute	Type	Mult.	Kind	Note
eventSource	VariableAccess	0..1	ref	The referenced VariableAccess raises this DataWrite CompletedEvent when the implicit write access was successful or an error occurred.

Table A.257: DataWriteCompletedEvent

Class	DcmIPdu			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Represents the IPdus handled by Dcm. **Tags:**atp.recommendedPackage=Pdus			
Base	ARObject, CollectableElement, FibexElement, IPdu, Identifiable, MultilanguageReferrable, Packageable Element, Pdu, Referrable			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
diagPduType	DiagPduType	1	attr	Attribute is used to distinguish a request from a response.

Table A.258: DcmIPdu

Class	DelegatedPortAnnotation			
Package	M2::AUTOSARTemplates::SWComponentTemplate::ApplicationAttributes			
Note	Annotation to a "delegated port" to specify the Signal Fan In or Signal Fan Out inside the CompositionSwComponentType.			
Base	*ARObject*, *GeneralAnnotation*			
Aggregated by	*PortPrototype*.delegatedPortAnnotation			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
signalFan	SignalFanEnum	0..1	attr	Specifies the Signal Fan In or Signal Fan Out inside the Composition Type.

Table A.259: DelegatedPortAnnotation

Class	DelegationSwConnector			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Composition			
Note	A delegation connector delegates one inner PortPrototype (a port of a component that is used inside the composition) to a outer PortPrototype of compatible type that belongs directly to the composition (a port that is owned by the composition).			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *SwConnector*			
Aggregated by	*AtpClassifier*.atpFeature, CompositionSwComponentType.connector			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
innerPort	PortPrototype	0..1	iref	The port that belongs to the ComponentPrototype in the composition **Tags:**xml.typeElement=true **InstanceRef implemented by:**PortInCompositionType InstanceRef
outerPort	PortPrototype	0..1	ref	The port that is located on the outside of the Composition Type

Table A.260: DelegationSwConnector

Class	DependencyOnArtifact			
Package	M2::AUTOSARTemplates::CommonStructure::Implementation			
Note	Dependency on the existence of another artifact, e.g. a library.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*Implementation*.generatedArtifact, *Implementation*.requiredArtifact, *Implementation*.requiredGeneratorTool			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
artifact Descriptor	AutosarEngineering Object	0..1	aggr	The specified artifact needs to exist.
usage	DependencyUsage Enum	*	attr	Specification for which process step(s) this dependency is required.

Table A.261: DependencyOnArtifact

Class	Dhcpv6Props
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology
Note	This meta-class specifies the configuration options for DHCPv6.
Base	*ARObject*
Aggregated by	Ipv6Props.dhcpProps

▽

Document ID 635: AUTOSAR_TR_ClassicAutosarModelConstraints

△

Class	Dhcpv6Props			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
tcpIpDhcp V6CnfDelayMax	TimeValue	0..1	attr	Maximum delay in seconds before sending the first Confirm message. If this value is bigger than the previous minimum delay value a random delay will be chosen from the interval.
tcpIpDhcp V6CnfDelayMin	TimeValue	0..1	attr	Minimum delay in seconds before the first Confirm message will be sent.
tcpIpDhcpV6Inf DelayMax	TimeValue	0..1	attr	Maximum delay in seconds before sending the first Information Request message. If this value is bigger than the previous minimum delay value a random delay will be chosen from the interval.
tcpIpDhcpV6Inf DelayMin	TimeValue	0..1	attr	Minimum delay (s) before the first Information Request message will be sent.
tcpIpDhcpV6Sol DelayMax	TimeValue	0..1	attr	Maximum delay in seconds before sending the first Solicit message. If this value is bigger than the previous minimum delay value a random delay will be chosen from the interval.
tcpIpDhcpV6Sol DelayMin	TimeValue	0..1	attr	Minimum delay (s) before the first Solicit message will be sent.

Table A.262: Dhcpv6Props

Class	DiagEventDebounceCounterBased			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	This meta-class represents the ability to indicate that the counter-based debounce algorithm shall be used by the DEM for this diagnostic monitor. This is related to set the ECUC choice container DemDebounceAlgorithmClass to DemDebounceCounterBased.			
Base	ARObject, DiagEventDebounceAlgorithm, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	DiagnosticDebounceAlgorithmProps.debounceAlgorithm, DiagnosticEventNeeds.diagEventDebounceAlgorithm			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
counterBased FdcThreshold StorageValue	Integer	0..1	attr	Threshold to allocate an event memory entry and to capture the Freeze Frame.
counter DecrementStep Size	Integer	0..1	attr	This value shall be taken to decrement the internal debounce counter. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime
counterFailed Threshold	Integer	0..1	attr	This value defines the event-specific limit that indicates the "failed" counter status. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime
counter IncrementStep Size	Integer	0..1	attr	This value shall be taken to increment the internal debounce counter. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime
counterJump Down	Boolean	0..1	attr	This value activates or deactivates the counter jump-down behavior. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime

▽

    Document ID 635: AUTOSAR_TR_ClassicAutosarModelConstraints

△

Class	DiagEventDebounceCounterBased			
counterJump DownValue	Integer	0..1	attr	This value represents the initial value of the internal debounce counter if the counting direction changes from incrementing to decrementing. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime
counterJumpUp	Boolean	0..1	attr	This value activates or deactivates the counter jump-up behavior. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime
counterJumpUp Value	Integer	0..1	attr	This value represents the initial value of the internal debounce counter if the counting direction changes from decrementing to incrementing. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime
counterPassed Threshold	Integer	0..1	attr	This value defines the event-specific limit that indicates the "passed" counter status. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime

Table A.263: DiagEventDebounceCounterBased

Class	DiagEventDebounceMonitorInternal			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	"This meta-class represents the ability to indicate that no Dem pre-debounce algorithm shall be used for this diagnostic monitor. The SWC might implement an internal debouncing algorithm and report qualified (debounced) results to the Dem/DM.			
Base	*ARObject*, *DiagEventDebounceAlgorithm*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DiagnosticDebounceAlgorithmProps.debounceAlgorithm, DiagnosticEventNeeds.diagEventDebounce Algorithm			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.264: DiagEventDebounceMonitorInternal

Class	DiagEventDebounceTimeBased			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	This meta-class represents the ability to indicate that the time-based pre-debounce algorithm shall be used by the Dem for this diagnostic monitor. This is related to set the EcuC choice container DemDebounceAlgorithmClass to DemDebounceTime Base.			
Base	*ARObject*, *DiagEventDebounceAlgorithm*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DiagnosticDebounceAlgorithmProps.debounceAlgorithm, DiagnosticEventNeeds.diagEventDebounce Algorithm			
Attribute	Type	Mult.	Kind	Note
timeBasedFdc Threshold StorageValue	TimeValue	0..1	attr	Threshold to allocate an event memory entry and to capture the Freeze Frame. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime

▽

△

Class	DiagEventDebounceTimeBased			
timeFailed Threshold	TimeValue	0..1	attr	This value represents the event-specific delay indicating the "failed" status.
				Stereotypes: atpVariation **Tags:**vh.latestBindingTime=preCompileTime
timePassed Threshold	TimeValue	0..1	attr	This value represents the event-specific delay indicating the "passed" status.
				Stereotypes: atpVariation **Tags:**vh.latestBindingTime=preCompileTime

Table A.265: DiagEventDebounceTimeBased

Class	*DiagnosticAbstractDataIdentifier* (abstract)			
Package	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
Note	This meta-class represents an abstract base class for the modeling of a diagnostic data identifier (DID).			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Subclasses	DiagnosticDataIdentifier, DiagnosticDynamicDataIdentifier			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
id	PositiveInteger	0..1	attr	This is the numerical identifier used to identify the DiagnosticAbstractDataIdentifier in the scope of diagnostic workflow
				Stereotypes: atpVariation **Tags:**vh.latestBindingTime=preCompileTime

Table A.266: DiagnosticAbstractDataIdentifier

Class	*DiagnosticAbstractParameter* (abstract)			
Package	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
Note	This meta-class represents an abstract base class for modeling a diagnostic parameter.			
Base	*ARObject*			
Subclasses	DiagnosticParameter, DiagnosticParameterElement			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
bitOffset	PositiveInteger	0..1	attr	This represents the bitOffset of the DiagnosticParameter. The value of the bitOffset shall always be interpreted as relative to the start of the enclosing DiagnosticData Identifier, DiagnosticParameterIdentifier, or Diagnostic RoutineSubfunction.
				Stereotypes: atpIdentityContributor **Tags:**atp.Status=candidate
dataElement	DiagnosticDataElement	0..1	aggr	This represents the related dataElement of the Diagnostic Parameter
				Stereotypes: atpSplitable; atpVariation **Tags:** atp.Splitkey=dataElement.shortName, data Element.variationPoint.shortLabel vh.latestBindingTime=postBuild

▽

Document ID 635: AUTOSAR_TR_ClassicAutosarModelConstraints

△

Class	DiagnosticAbstractParameter (abstract)			
parameterSize	PositiveInteger	0..1	attr	This attribute allows for the specification of the parameter size. This information is relevant if there is a gap between one diagnostic parameter and the following diagnostic parameter (or the tail of the telegram). The unit is bit and the values shall be multiples of 8. **Tags:**atp.Status=candidate

Table A.267: DiagnosticAbstractParameter

Class	DiagnosticAccessPermission			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm			
Note	This represents the specification of whether a given service can be accessed according to the existence of meta-classes referenced by a particular DiagnosticAccessPermission. In other words, this meta-class acts as a mapping element between several (otherwise unrelated) pieces of information that are put into context for the purpose of checking for access rights. **Tags:**atp.recommendedPackage=DiagnosticAccessPermissions			
Base	_ARElement_, _ARObject_, _CollectableElement_, _DiagnosticCommonElement_, _Identifiable_, _Multilanguage Referrable_, _PackageableElement_, _Referrable_			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
authentication Role	DiagnosticAuthRole	*	ref	This reference identifies the authenticationRole applicable for the enclosing DiagnosticMemoryDestinationUser Defined.
diagnostic Session	DiagnosticSession	*	ref	This represents the associated DiagnosticSessions
environmental Condition	Diagnostic EnvironmentalCondition	0..1	ref	This represents the environmental conditions associated with the access permission.
securityLevel	DiagnosticSecurityLevel	*	ref	This represents the associated DiagnosticSecurityLevels

Table A.268: DiagnosticAccessPermission

Class	DiagnosticAging			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticAging			
Note	Defines the aging algorithm. **Tags:**atp.recommendedPackage=DiagnosticAgings			
Base	_ARElement_, _ARObject_, _CollectableElement_, _DiagnosticCommonElement_, _Identifiable_, _Multilanguage Referrable_, _PackageableElement_, _Referrable_			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
agingCycle	DiagnosticOperation Cycle	0..1	ref	This represents the applicable aging cycle. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=agingCycle.diagnosticOperationCycle, aging Cycle.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
threshold	PositiveInteger	0..1	attr	Number of aging cycles needed to unlearn/delete the event. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime

Table A.269: DiagnosticAging

Class	*DiagnosticAuthentication* (abstract)			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::Authentication			
Note	This meta-class represents the ability to configure the usage of the UDS service Authentication in the Diagnostic extract.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	DiagnosticAuthenticationConfiguration, DiagnosticDeAuthentication, DiagnosticProofOfOwnership, DiagnosticVerifyCertificateBidirectional, DiagnosticVerifyCertificateUnidirectional			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
authentication Class	Diagnostic AuthenticationClass	0..1	ref	This represents the corresponding "class", i.e. this meta-class provides properties that are shared among all instances of applicable sub-classes of DiagnosticService Instance. The subclasses that affected by this pattern implement references to the applicable "class"-role that substantiate this abstract reference.
authentication Timeout	TimeValue	0..1	attr	This attribute defines the time that the authentication state is maintained in default-session if there is no communication from the authenticated client.

Table A.270: DiagnosticAuthentication

Class	**DiagnosticAuthenticationConfiguration**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::Authentication			
Note	This meta-class represents the subfunction to configure the authentication. **Tags:**atp.recommendedPackage=DiagnosticAuthentications			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticAuthentication*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.271: DiagnosticAuthenticationConfiguration

Class	**DiagnosticClearResetEmissionRelatedInfo**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x04_ClearResetEmission RelatedInfo			
Note	This meta-class represents the ability to model an instance of the OBD mode 0x04 service. **Tags:**atp.recommendedPackage=DiagnosticClearResetEmissionRelatedInfos			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
clearReset Emission Related DiagnosticInfo Class	DiagnosticClearReset EmissionRelatedInfo Class	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticClearReste EmissionRelatedInfo in the given context.

Table A.272: DiagnosticClearResetEmissionRelatedInfo

Class	DiagnosticComControl			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::CommunicationControl			
Note	This represents an instance of the "Communication Control" diagnostic service. **Tags:**atp.recommendedPackage=DiagnosticCommunicationControls			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
comControl Class	DiagnosticComControl Class	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticComControl in the given context.
customSub Function Number	PositiveInteger	0..1	attr	This attribute shall be used to define a custom sub-function number if none of the standardized values of category shall be used.

Table A.273: DiagnosticComControl

Class	DiagnosticComControlSpecificChannel			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::CommunicationControl			
Note	This represents the ability to add further attributes to the definition of a specific channel that is subject to the diagnostic service "communication control".			
Base	*ARObject*			
Aggregated by	DiagnosticComControlClass.specificChannel			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
specificChannel	CommunicationCluster	0..1	ref	This represents the affected CommunicationClusters in the role specificChannel
subnetNumber	PositiveInteger	0..1	attr	This represents the applicable subnet number (which is an arbitrary number ranging from 1..14)

Table A.274: DiagnosticComControlSpecificChannel

Class	DiagnosticComControlSubNodeChannel			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::CommunicationControl			
Note	This represents the ability to add further attributes to the definition of a specific sub-node channel that is subject to the diagnostic service "communication control".			
Base	*ARObject*			
Aggregated by	DiagnosticComControlClass.subNodeChannel			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
subNode Channel	CommunicationCluster	0..1	ref	This represents the affected CommunicationClusters in the role subNodeChannel
subNode Number	PositiveInteger	0..1	attr	This represents the applicable subNode number. The value corresponds to the request message parameter nodeIdentificationNumber of diagnostic service CommunicationControl (0x28).

Table A.275: DiagnosticComControlSubNodeChannel

Class	DiagnosticCommonElement (abstract)			
Package	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
Note	This meta-class represents a common base class for all diagnostic elements. It does not contribute any specific functionality other than the ability to become the target of a reference.			
Base	ARElement, ARObject, CollectableElement, Identifiable, MultilanguageReferrable, Packageable Element, Referrable			
Subclasses	DiagnosticAbstractAliasEvent, DiagnosticAbstractDataIdentifier, DiagnosticAccessPermission, DiagnosticAging, DiagnosticAuthRole, DiagnosticCondition, DiagnosticConditionGroup, Diagnostic CustomServiceClass, DiagnosticDataIdentifierSet, DiagnosticEcuInstanceProps, Diagnostic EnvironmentalCondition, DiagnosticEvent, DiagnosticExtendedDataRecord, DiagnosticFimEventGroup, DiagnosticFreezeFrame, DiagnosticFunctionIdentifier, DiagnosticFunctionIdentifierInhibit, Diagnostic Indicator, DiagnosticInfoType, DiagnosticIumpr, DiagnosticIumprDenominatorGroup, DiagnosticIumpr Group, DiagnosticJ1939ExpandedFreezeFrame, DiagnosticJ1939FreezeFrame, DiagnosticJ1939Node, DiagnosticJ1939Spn, DiagnosticMapping, DiagnosticMeasurementIdentifier, DiagnosticMemory Destination, DiagnosticMemoryIdentifier, DiagnosticOperationCycle, DiagnosticParameterIdentifier, DiagnosticPowertrainFreezeFrame, DiagnosticProtocol, DiagnosticRoutine, DiagnosticSecurityLevel, DiagnosticServiceClass, DiagnosticServiceInstance, DiagnosticServiceTable, DiagnosticSession, DiagnosticTestResult, DiagnosticTestRoutineIdentifier, DiagnosticTroubleCode, DiagnosticTroubleCode Group, DiagnosticTroubleCodeProps			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.276: DiagnosticCommonElement

Class	<<atpVariation>> DiagnosticCommonProps			
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticCommonProps			
Note	This meta-class aggregates a number of common properties that are shared among a diagnostic extract. **Tags:**vh.latestBindingTime=codeGenerationTime			
Base	ARObject			
Aggregated by	DiagnosticContributionSet.commonProperties			
Attribute	Type	Mult.	Kind	Note
debounce AlgorithmProps	DiagnosticDebounce AlgorithmProps	*	aggr	Defines the used debounce algorithms relevant in the context of the enclosing DiagnosticCommonProps. Usually, there is a variety of debouncing algorithms to take into account and therefore the multiplicity of this aggregation is set to 0..*.
default Endianness	ByteOrderEnum	0..1	attr	Defines the default endianness of the data belonging to a DID or RID which is applicable if the DiagnosticData Element does not define the endianness via the swData DefProps.baseType attribute.
event Combination Reporting Behavior	DiagnosticEvent CombinationReporting BehaviorEnum	0..1	attr	In case of EventCombination on Retrieval, this attribute specifies if a specific order of reporting is to be maintained.
maxNumberOf Request Correctly Received Response Pending	PositiveInteger	0..1	attr	Maximum number of negative responses with response code 0x78 (requestCorrectlyReceived-ResponsePending) allowed per request. DCM will send a negative response with response code 0x10 (generalReject), in case the limit value gets reached. Value 0xFF means that no limit number of NRC 0x78 response apply.
occurrence Counter Processing	DiagnosticOccurrence CounterProcessing Enum	0..1	attr	This attribute defines the consideration of the fault confirmation process for the occurrence counter.
resetConfirmed BitOnOverflow	Boolean	0..1	attr	This attribute defines, whether the confirmed bit is reset or not while an event memory entry will be displaced.

▽

△

Class	<<atpVariation>> **DiagnosticCommonProps**			
responseOnAll RequestSids	Boolean	0..1	attr	If set to FALSE the DCM will not respond to diagnostic request that contains a service ID which is in the range from 0x40 to 0x7F or in the range from 0xC0 to 0xFF (Response IDs).
responseOn Second Declined Request	Boolean	0..1	attr	Defines the reaction upon a second request (ClientB) that can not be processed (e.g. due to priority assessment). TRUE: when the second request (Client B) can not be processed, it shall be answered with NRC21 BusyRepeat Request. FALSE: when the second request (Client B) can not be processed, it shall not be responded.
typeOfEvent Combination Supported	DiagnosticEvent CombinationBehavior Enum	0..1	attr	Select type of Event Combination support.

Table A.277: DiagnosticCommonProps

Enumeration	**DiagnosticCompareTypeEnum**
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::EnvironmentalCondition
Note	Enumeration for the type of a comparison of values usually expressed by the following operators: ==, !=, <, <=, >, >=
Aggregated by	*DiagnosticEnvCompareCondition*.compareType
Literal	**Description**
isEqual	equal **Tags:**atp.EnumerationLiteralIndex=0
isGreaterOrEqual	greater than or equal **Tags:**atp.EnumerationLiteralIndex=5
isGreaterThan	greater than **Tags:**atp.EnumerationLiteralIndex=4
isLessOrEqual	less than or equal **Tags:**atp.EnumerationLiteralIndex=3
isLessThan	less than **Tags:**atp.EnumerationLiteralIndex=2
isNotEqual	not equal **Tags:**atp.EnumerationLiteralIndex=1

Table A.278: DiagnosticCompareTypeEnum

Class	*DiagnosticCondition* (abstract)			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticCondition			
Note	Abstract element for StorageConditions and EnableConditions.			
Base	*ARElement*, ARObject, CollectableElement, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Subclasses	DiagnosticEnableCondition, DiagnosticStorageCondition			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	DiagnosticCondition (abstract)			
initValue	Boolean	0..1	attr	Defines the initial status for enable or disable of acceptance/storage of event reports of a diagnostic event. The value is the initialization after power up (before this condition is reported the first time). true: acceptance/storage of a diagnostic event enabled false: acceptance/storage of a diagnostic event disabled

Table A.279: DiagnosticCondition

Class	DiagnosticConnectedIndicator			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticEvent			
Note	Description of indicators that are defined per DiagnosticEvent.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	DiagnosticEvent.connectedIndicator			
Attribute	Type	Mult.	Kind	Note
behavior	DiagnosticConnected IndicatorBehaviorEnum	0..1	attr	Behavior of the linked indicator.
healingCycle	DiagnosticOperation Cycle	0..1	ref	The deactivation of indicators per event is defined as healing of a diagnostic event. The operation cycle in which the warning indicator will be switched off is defined here.
healingCycle Counter Threshold	PositiveInteger	0..1	attr	This attribute defines the number of healing cycles for the WarningIndicatorOffCriteria **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime
indicator	DiagnosticIndicator	0..1	ref	Reference to the used indicator.
indicatorFailure CycleCounter Threshold	PositiveInteger	0..1	attr	This attribute defines the number of failure cycles for the WarningIndicatorOnCriteria. Please note that this attribute is not relevant for the Adaptive Platform.

Table A.280: DiagnosticConnectedIndicator

Document ID 635: AUTOSAR_TR_ClassicAutosarModelConstraints

Class	DiagnosticConnection			
Package	M2::AUTOSARTemplates::SystemTemplate::DiagnosticConnection			
Note	DiagnosticConncection that is used to describe the relationship between several TP connections. **Tags:**atp.recommendedPackage=DiagnosticConnections			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
functional Request	TpConnectionIdent	*	ref	Reference to functional request messages.
periodic ResponseUudt	PduTriggering	*	ref	Reference to UUDT responses.
physical Request	TpConnectionIdent	0..1	ref	Reference to a physical request message.
response	TpConnectionIdent	0..1	ref	In the vast majority of cases a response is required. However, there are also cases where providing the response is not possible and/or not allowed.
responseOn Event	TpConnectionIdent	0..1	ref	Reference to a ROE message. **Tags:**atp.Status=obsolete

Table A.281: DiagnosticConnection

Class	DiagnosticContributionSet			
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticContribution			
Note	This meta-class represents a root node of a diagnostic extract. It bundles a given set of diagnostic model elements. The granularity of the DiagonsticContributionSet is arbitrary in order to support the aspect of decentralized configuration, i.e. different contributors can come up with an own DiagnosticContribution Set. **Tags:**atp.recommendedPackage=DiagnosticContributionSets			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
common Properties	DiagnosticCommon Props	0..1	aggr	This attribute represents a collection of diagnostic properties that are shared among the entire Diagnostic ContributionSet. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=commonProperties
element	DiagnosticCommon Element	*	ref	This represents a DiagnosticCommonElement considered in the context of the DiagnosticContributionSet **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=element.diagnosticCommonElement, element.variationPoint.shortLabel vh.latestBindingTime=postBuild
serviceTable	DiagnosticServiceTable	*	ref	This represents the collection of DiagnosticServiceTables to be considered in the scope of this Diagnostic ContributionSet. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=serviceTable.diagnosticServiceTable, service Table.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.282: DiagnosticContributionSet

Class	DiagnosticControlEnableMaskBit			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::IOControl			
Note	This meta-class has the ability to represent one bit in the control enable mask record.			
Base	ARObject			
Aggregated by	DiagnosticIOControl.controlEnableMaskBit			
Attribute	Type	Mult.	Kind	Note
bitNumber	PositiveInteger	0..1	attr	This attribute represents the bit number of the bit in the control mask record. Bit number 0 is the most significant bit (MSB) in the first byte of the CEMR in the network presentation.
controlledData Element	DiagnosticDataElement	*	ref	This reference represents the collection of Diagnostic DataElements that are controlled by this bit of the control mask record.

Table A.283: DiagnosticControlEnableMaskBit

Class	DiagnosticCustomServiceClass			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::CommonService			
Note	This represents the ability to define a custom diagnostic service class and assign an ID to it. Further configuration is not foreseen from the point of view of the diagnostic extract and consequently needs to be done on the level of ECUC. **Tags:**atp.recommendedPackage=DiagnosticCustomServiceClasses			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceClass, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
customService Id	PositiveInteger	0..1	attr	This attribute may only be used for the definition of custom services. The values shall not overlap with existing standardized service IDs.

Table A.284: DiagnosticCustomServiceClass

Class	DiagnosticDataByIdentifier (abstract)			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::DataByIdentifier			
Note	This represents an abstract base class for all diagnostic services that access data by identifier.			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Subclasses	DiagnosticReadDataByIdentifier, DiagnosticReadScalingDataByIdentifier, DiagnosticWriteDataBy Identifier			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
dataIdentifier	DiagnosticAbstractData Identifier	0..1	ref	This represents the linked DiagnosticDataIdentifier.

Table A.285: DiagnosticDataByIdentifier

Class	DiagnosticDataElement
Package	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics
Note	This meta-class represents the ability to describe a concrete piece of data to be taken into account for diagnostic purposes.
Base	ARObject, DiagnosticServiceMappingDiagTarget, Identifiable, MultilanguageReferrable, Referrable

▽

Document ID 635: AUTOSAR_TR_ClassicAutosarModelConstraints

△

Class	DiagnosticDataElement			
Aggregated by	*DiagnosticAbstractParameter*.dataElement			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
arraySize Semantics	ArraySizeSemantics Enum	0..1	attr	This attribute controls the meaning of the value of the array size.
maxNumberOf Elements	PositiveInteger	0..1	attr	The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.
scalingInfoSize	PositiveInteger	0..1	attr	Size in bytes of scaling information for the DiagnosticData Element if used with DiagnosticReadScalingDataBy Identifier
swDataDef Props	SwDataDefProps	0..1	aggr	This property allows to specify data definition properties in order to support the definition of e.g. computation formulae and data constraints. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=swDataDefProps

Table A.286: DiagnosticDataElement

Class	DiagnosticDataIdentifier			
Package	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
Note	This meta-class represents the ability to model a diagnostic data identifier (DID) that is fully specified regarding the payload at configuration-time. **Tags:**atp.recommendedPackage=DiagnosticDataIdentifiers			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticAbstractDataIdentifier*, *DiagnosticCommon Element*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dataElement	DiagnosticParameter	*	aggr	This is the dataElement associated with the Diagnostic DataIdentifier. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=dataElement.bitOffset, data Element.ident.shortName, dataElement.variation Point.shortLabel vh.latestBindingTime=postBuild
didSize	PositiveInteger	0..1	attr	This attribute indicates the size in bytes of the Diagnostic DataIdentifier.
representsVin	Boolean	0..1	attr	This attributes indicates whether the specific Diagnostic DataIdentifier represents the vehicle identification.
supportInfoByte	DiagnosticSupportInfo Byte	0..1	aggr	This attribute represents the supported information associated with the DiagnosticDataIdentifier.

Table A.287: DiagnosticDataIdentifier

Class	DiagnosticDataIdentifierSet
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode
Note	This represents the ability to define a list of DiagnosticDataIdentifiers that can be reused in different contexts. **Tags:**atp.recommendedPackage=DiagnosticDataIdentifierSets

▽

△

Class	DiagnosticDataIdentifierSet			
Base	*ARElement*, ARObject, CollectableElement, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dataIdentifier (ordered)	DiagnosticDataIdentifier	*	ref	Reference to an ordered list of Data Identifiers.

Table A.288: DiagnosticDataIdentifierSet

Class	DiagnosticDeAuthentication			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::Authentication			
Note	This meta-class represents the subfunction to remove the authentication **Tags:**atp.recommendedPackage=DiagnosticAuthentications			
Base	*ARElement*, ARObject, CollectableElement, *DiagnosticAuthentication*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.289: DiagnosticDeAuthentication

Class	DiagnosticDebounceAlgorithmProps			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticDebouncingAlgorithm			
Note	Defines properties for the debounce algorithm class.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DiagnosticCommonProps.debounceAlgorithmProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
debounce Algorithm	DiagEventDebounce Algorithm	0..1	aggr	This represents the actual debounce algorithm.
debounce Behavior	DiagnosticDebounce BehaviorEnum	0..1	attr	This attribute defines how the event debounce algorithm will behave, if a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime
debounce CounterStorage	Boolean	0..1	attr	Switch to store the debounce counter value non-volatile or not. true: debounce counter value shall be stored non-volatile false: debounce counter value is volatile Please note that this attribute is not relevant for the adaptive platform.

Table A.290: DiagnosticDebounceAlgorithmProps

Class	DiagnosticDemProvidedDataMapping			
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticMapping::ServiceMapping			
Note	This represents the ability to define the nature of a data access for a DiagnosticDataElement in the Dem. **Tags:**atp.recommendedPackage=DiagnosticServiceMappings			
Base	*ARElement*, ARObject, CollectableElement, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dataElement	DiagnosticDataElement	0..1	ref	This represents the DiagnosticDataElement for which the access is further qualified by the DiagnosticDemProvided DataMapping.
dataProvider	NameToken	0..1	attr	This represents the ability to further specify the access within the Dem.

Table A.291: DiagnosticDemProvidedDataMapping

Class	DiagnosticDynamicDataIdentifier			
Package	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
Note	This meta-class represents the ability to define a diagnostic data identifier (DID) at run-time. **Tags:**atp.recommendedPackage=DiagnosticDataIdentifiers			
Base	*ARElement*, ARObject, CollectableElement, *DiagnosticAbstractDataIdentifier*, *DiagnosticCommon Element*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.292: DiagnosticDynamicDataIdentifier

Class	DiagnosticDynamicallyDefineDataIdentifier			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::DynamicallyDefineDataIdentifier			
Note	This represents an instance of the "Dynamically Define Data Identifier" diagnostic service. **Tags:**atp.recommendedPackage=DiagnosticDynamicallyDefineDataIdentifiers			
Base	*ARElement*, ARObject, CollectableElement, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dataIdentifier	DiagnosticDynamicData Identifier	0..1	ref	This represents the applicable DiagnosticDynamicData Identfier.
dynamically DefineData IdentifierClass	DiagnosticDynamically DefineDataIdentifier Class	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticDynamicallyDefine DataIdentifier in the given context.
maxSource Element	PositiveInteger	0..1	attr	This represents the maximum number of source elements of the dynamically created DID.

Table A.293: DiagnosticDynamicallyDefineDataIdentifier

Class	DiagnosticDynamicallyDefineDataIdentifierClass
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::DynamicallyDefineDataIdentifier
Note	This meta-class contains attributes shared by all instances of the "Dynamically Define Data Identifier" diagnostic service. **Tags:**atp.recommendedPackage=DiagnosticDynamicallyDefineDataIdentifiers
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*
Aggregated by	ARPackage.element

Attribute	Type	Mult.	Kind	Note
checkPer SourceId	Boolean	0..1	attr	If set to TRUE, the Dcm module shall check the session, security and mode dependencies per source DIDs with a ReadDataByIdentifier (0x22) with DID in the range 0x F200 to 0xF3FF. If set to FALSE. the Dcm module shall not check the session, security and mode dependencies per source DIDs with a ReadDataByIdentifier (0x22) with DID in the range 0xF200 to 0xF3FF.
configuration Handling	DiagnosticHandleDDDI ConfigurationEnum	0..1	attr	This configuration switch defines whether DDDID definition is handled as non-volatile information or not.
subfunction	DiagnosticDynamically DefineDataIdentifier SubfunctionEnum	*	attr	This attribute contains a list of applicable subfunctions for all DiagnosticDynamicallyDefineDataIdentifier that reference the DiagnosticDynamicallyDefineDataIdentifier Class.

Table A.294: DiagnosticDynamicallyDefineDataIdentifierClass

Enumeration	DiagnosticDynamicallyDefineDataIdentifierSubfunctionEnum
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::DynamicallyDefineData Identifier
Note	This meta-class contains a list of possible subfunctions for the UDS service 0x2C.
Aggregated by	DiagnosticDynamicallyDefineDataIdentifierClass.subfunction

Literal	Description
clearDynamically DefineDataIdentifier	Clear the specified dynamic data identifier. **Tags:**atp.EnumerationLiteralIndex=0
defineByIdentifier	The definition of dynamic data identifier shall be done via a reference to a diagnostic data identifier. **Tags:**atp.EnumerationLiteralIndex=1
defineByMemory Address	The definition of dynamic data identifier shall be done via a reference to a memory address. **Tags:**atp.EnumerationLiteralIndex=2

Table A.295: DiagnosticDynamicallyDefineDataIdentifierSubfunctionEnum

Class	DiagnosticEcuInstanceProps
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticContribution
Note	This meta-class represents the ability to model properties that are specific for a given EcuInstance but on the other hand represent purely diagnostic-related information. In the spirit of decentralized configuration it is therefore possible to specify the diagnostic-related information related to a given EcuInstance even if the EcuInstance does not yet exist. **Tags:**atp.recommendedPackage=DiagnosticEcuInstancePropss
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*
Aggregated by	ARPackage.element

▽

△

Class	DiagnosticEcuInstanceProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
ecuInstance	EcuInstance	*	ref	This represents the actual EcuInstance to which the information contained in the DiagnosticEcuInstance contribute. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=ecuInstance
obdSupport	DiagnosticObdSupport Enum	0..1	attr	This attribute is used to specify the role (if applicable) in which the DiagnosticEcuInstance supports OBD.

Table A.296: DiagnosticEcuInstanceProps

Class	DiagnosticEcuReset			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::EcuReset			
Note	This represents an instance of the "ECU Reset" diagnostic service. **Tags:**atp.recommendedPackage=DiagnosticEcuResets			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
customSub Function Number	PositiveInteger	0..1	attr	This attribute shall be used to define a custom sub-function number if none of the standardized values of category shall be used.
ecuResetClass	DiagnosticEcuReset Class	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticEcuReset in the given context.

Table A.297: DiagnosticEcuReset

Class	DiagnosticEnableCondition			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticCondition			
Note	Specification of an enable condition. **Tags:**atp.recommendedPackage=DiagnosticConditions			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticCondition*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.298: DiagnosticEnableCondition

Class	DiagnosticEnableConditionGroup			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticConditionGroup			
Note	Enable condition group which includes one or several enable conditions. **Tags:**atp.recommendedPackage=DiagnosticConditions			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticConditionGroup*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			

▽

△

Class	DiagnosticEnableConditionGroup			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
enableCondition	DiagnosticEnable Condition	1..*	ref	Reference to enableConditions that are part of the Enable ConditionGroup. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=enableCondition.diagnosticEnableCondition, enableCondition.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.299: DiagnosticEnableConditionGroup

Class	DiagnosticEnableConditionPortMapping			
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticMapping			
Note	Defines to which SWC service ports with DiagnosticEnableConditionNeeds the DiagnosticEnable Condition is mapped. **Tags:**atp.recommendedPackage=DiagnosticMappings			
Base	*ARElement*, ARObject, CollectableElement, *DiagnosticCommonElement*, DiagnosticMapping, DiagnosticSwMapping, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
enableCondition	DiagnosticEnable Condition	0..1	ref	Reference to the EnableCondition which is mapped to a SWC service port with DiagnosticEnableConditionNeeds.
swcFlatService Dependency	SwcService Dependency	0..1	ref	Reference to a SwcServiceDependencyType that links ServiceNeeds to SWC service ports. This reference can be used in early stages of the development in order to identify the SwcServiceDependency without a full System Context.
swcService DependencyIn System	SwcService Dependency	0..1	iref	Instance reference to a SwcServiceDependency that links ServiceNeeds to SWC service ports. **InstanceRef implemented by:**SwcServiceDependency InSystemInstanceRef

Table A.300: DiagnosticEnableConditionPortMapping

Class	DiagnosticEnvBswModeElement			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::EnvironmentalCondition			
Note	This meta-class represents the ability to refer to a specific ModeDeclaration in the scope of a BswModule Description.			
Base	*ARObject*, *DiagnosticEnvModeElement*, *Referrable*			
Aggregated by	DiagnosticEnvironmentalCondition.modeElement			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
mode	ModeDeclaration	1	iref	This reference identifies both the ModeDeclarationGroup Prototype and the ModeDeclaration for the specific mode comparison. **InstanceRef implemented by:**ModeInBswModule DescriptionInstanceRef

Table A.301: DiagnosticEnvBswModeElement

Class	DiagnosticEnvCompareCondition (abstract)			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::EnvironmentalCondition			
Note	DiagnosticCompareConditions are atomic conditions. They are based on the idea of a comparison at runtime of some variable data with something constant. The type of the comparison (==, !=, <, <=, ...) is specified in DiagnosticCompareCondition.compareType.			
Base	ARObject, DiagnosticEnvConditionFormulaPart			
Subclasses	DiagnosticEnvDataCondition, DiagnosticEnvDataElementCondition, DiagnosticEnvModeCondition			
Aggregated by	DiagnosticEnvConditionFormula.part			
Attribute	Type	Mult.	Kind	Note
compareType	DiagnosticCompare TypeEnum	0..1	attr	This attributes represents the concrete type of the comparison.

Table A.302: DiagnosticEnvCompareCondition

Class	DiagnosticEnvConditionFormula			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::EnvironmentalCondition			
Note	A DiagnosticEnvConditionFormula embodies the computation instruction that is to be evaluated at runtime to determine if the DiagnosticEnvironmentalCondition is currently present (i.e. the formula is evaluated to true) or not (otherwise). The formula itself consists of parts which are combined by the logical operations specified by DiagnosticEnvConditionFormula.op.			
	If a diagnostic functionality cannot be executed because an environmental condition fails then the diagnostic stack shall send a negative response code (NRC) back to the client. The value of the NRC is directly related to the specific formula and is therefore formalized in the attribute DiagnosticEnvCondition Formula.nrcValue.			
Base	ARObject, DiagnosticEnvConditionFormulaPart			
Aggregated by	DiagnosticEnvConditionFormula.part, DiagnosticEnvironmentalCondition.formula			
Attribute	Type	Mult.	Kind	Note
nrcValue	PositiveInteger	0..1	attr	This attribute represents the concrete NRC value that shall be returned if the condition fails.
op	DiagnosticLogical OperatorEnum	0..1	attr	This attribute represents the concrete operator (supported operators: and, or) of the condition formula.
part (ordered)	DiagnosticEnvCondition FormulaPart	*	aggr	This aggregation represents the collection of formula parts that can be combined by logical operators.

Table A.303: DiagnosticEnvConditionFormula

Class	DiagnosticEnvDataCondition			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::EnvironmentalCondition			
Note	A DiagnosticEnvDataCondition is an atomic condition that compares the current value of the referenced DiagnosticDataElement with a constant value defined by the ValueSpecification. All compareTypes are supported.			
Base	ARObject, DiagnosticEnvCompareCondition, DiagnosticEnvConditionFormulaPart			
Aggregated by	DiagnosticEnvConditionFormula.part			
Attribute	Type	Mult.	Kind	Note
compareValue	ValueSpecification	0..1	aggr	This attribute represents a fixed compare value taken to evaluate the compare condition.
dataElement	DiagnosticDataElement	0..1	ref	This reference represents the related diagnostic data element.

Table A.304: DiagnosticEnvDataCondition

Class	DiagnosticEnvDataElementCondition			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::EnvironmentalCondition			
Note	This meta-class represents the ability to formulate a diagnostic environment condition based on the value of a data element owned by the application software.			
Base	ARObject, DiagnosticEnvCompareCondition, DiagnosticEnvConditionFormulaPart			
Aggregated by	DiagnosticEnvConditionFormula.part			
Attribute	Type	Mult.	Kind	Note
compareValue	ValueSpecification	0..1	aggr	This aggregation represents the definition of the compare value against which the value taken from the application software shall be compared.
dataPrototype	DataPrototype	0..1	iref	This instanceRef represent the ability to access a data element owned by the application software on the AUTOSAR classic platform. **InstanceRef implemented by:**DataPrototypeInSystemInstanceRef
swDataDefProps	SwDataDefProps	0..1	aggr	Via this aggregation it is possible to describe the properties of the data that is obtained from the application for the environmental condition. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=swDataDefProps

Table A.305: DiagnosticEnvDataElementCondition

Class	DiagnosticEnvModeCondition			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::EnvironmentalCondition			
Note	DiagnosticEnvModeCondition are atomic condition based on the comparison of the active Mode Declaration in a ModeDeclarationGroupProtoype with the constant value of a ModeDeclaration. The formulation of this condition uses only one DiagnosticEnvElement, which contains enough information to deduce the variable part (i.e. the part that changes at runtime) as well as the constant part of the comparison. Only DiagnosticCompareTypeEnum.isEqual or DiagnosticCompareTypeEnum.isNotEqual are eligible values for DiagnosticAtomicCondition.compareType.			
Base	ARObject, DiagnosticEnvCompareCondition, DiagnosticEnvConditionFormulaPart			
Aggregated by	DiagnosticEnvConditionFormula.part			
Attribute	Type	Mult.	Kind	Note
modeElement	DiagnosticEnvMode Element	0..1	ref	This reference represents both the ModeDeclaration GroupPrototype and the ModeDeclaration relevant for the mode comparison.

Table A.306: DiagnosticEnvModeCondition

Class	DiagnosticEnvModeElement (abstract)
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::EnvironmentalCondition
Note	All ModeDeclarations that are referenced in a DiagnosticEnvModeCondition shall be defined as a DiagnosticEnvModeElement of this DiagnosticEnvironmentalCondition. This concept keeps the ARXML clean: It avoids that the DiagnosticEnvConditionFormula is cluttered by lengthy InstanceRef definitions. Furthermore, it allows that an InstanceRef only needs to be defined once and can be used multiple times in the different DiagnosticEnvModeConditions.
Base	ARObject, Referrable
Subclasses	DiagnosticEnvBswModeElement, DiagnosticEnvSwcModeElement

▽

△

Class	**DiagnosticEnvModeElement** (abstract)			
Aggregated by	DiagnosticEnvironmentalCondition.modeElement			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.307: DiagnosticEnvModeElement

Class	**DiagnosticEnvSwcModeElement**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::EnvironmentalCondition			
Note	This meta-class represents the ability to refer to a ModeDeclaration in a concrete System context.			
Base	*ARObject*, *DiagnosticEnvModeElement*, *Referrable*			
Aggregated by	DiagnosticEnvironmentalCondition.modeElement			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
mode	ModeDeclaration	0..1	iref	This reference identifies both the ModeDeclarationGroup Prototype and the ModeDeclaration for the specific mode comparison. **InstanceRef implemented by:**PModeInSystemInstance Ref

Table A.308: DiagnosticEnvSwcModeElement

Class	**DiagnosticEnvironmentalCondition**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::EnvironmentalCondition			
Note	The meta-class DiagnosticEnvironmentalCondition formalizes the idea of a condition which is evaluated during runtime of the ECU by looking at "environmental" states (e.g. one such condition is that the vehicle is not driving, i.e. vehicle speed == 0). **Tags:**atp.recommendedPackage=DiagnosticEnvironmentalConditions			
Base	*ARElement*, ARObject, CollectableElement, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
formula	DiagnosticEnvCondition Formula	0..1	aggr	This attribute represents the formula part of the DiagnosticEnvironmentalCondition.
modeElement	DiagnosticEnvMode Element	*	aggr	This aggregation contains a representation of Mode Declarations in the context of a DiagnosticEnvironmental Condition.

Table A.309: DiagnosticEnvironmentalCondition

Class	**DiagnosticEvent**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticEvent			
Note	This element is used to configure DiagnosticEvents. **Tags:**atp.recommendedPackage=DiagnosticEvents			
Base	*ARElement*, ARObject, CollectableElement, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	DiagnosticEvent			
associated Event Identification	PositiveInteger	0..1	attr	This attribute represents the identification number that is associated with the enclosing DiagnosticEvent and allows to identify it when placed into a snapshot record or extended data record storage. This value can be reported as internal data element in snapshot records or extended data records.
clearEvent Allowed Behavior	DiagnosticClearEvent AllowedBehaviorEnum	0..1	attr	This attribute defines the resulting UDS status byte for the related event, which shall not be cleared according to the ClearEventAllowed callback
confirmation Threshold	PositiveInteger	0..1	attr	This attribute defines the number of operation cycles with a failed result before a confirmed DTC is set to 1. The semantic of this attribute is a by "1" increased value compared to the confirmation threshold of the "trip counter" mentioned in ISO 14229-1 in figure D.4. A value of "1" defines the immediate confirmation of the DTC along with the first reported failed. This is also sometimes called "zero trip DTC". A value of "2" defines a DTC confirmation in the operation cycle after the first occurred failed. A value of "2" is typically used in the US for OBD DTC confirmation. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime
connected Indicator	DiagnosticConnected Indicator	*	aggr	Event specific description of Indicators. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=connectedIndicator.shortName, connected Indicator.variationPoint.shortLabel vh.latestBindingTime=postBuild
eventClear Allowed	DiagnosticEventClear AllowedEnum	0..1	attr	This attribute defines whether the Dem has access to a "ClearEventAllowed" callback.
eventKind	DiagnosticEventKind Enum	0..1	attr	This attribute is used to distinguish between SWC and BSW events.
prestorage FreezeFrame	Boolean	0..1	attr	This attribute describes whether the Prestorage of Freeze Frames is supported by the assigned event or not. True: Prestorage of FreezeFrames is supported False: Prestorage of FreezeFrames is not supported
prestored Freezeframe StoredInNvm	Boolean	0..1	attr	If the Event uses a prestored freeze-frame (using the operations PrestoreFreezeFrame and ClearPrestored FreezeFrame of the service interface DiagnosticMonitor) this attribute indicates if the Event requires the data to be stored in non-volatile memory. TRUE = Dem shall store the prestored data in non-volatile memory, FALSE = Data can be lost at shutdown (not stored in Nvm)
recoverableIn SameOperation Cycle	Boolean	0..1	attr	If the attribute is set to true then reporting PASSED will reset the indication of a failed test in the current operation cycle. If the attribute is set to false then reporting PASSED will be ignored and not lead to a reset of the indication of a failed test.

Table A.310: DiagnosticEvent

Enumeration	DiagnosticEventCombinationBehaviorEnum
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticCommonProps
Note	Select type of Event Combination support
Aggregated by	DiagnosticCommonProps.typeOfEventCombinationSupported

▽

△

Enumeration	DiagnosticEventCombinationBehaviorEnum
Literal	Description
eventCombination OnRetrieval	Event combination on retrieval is used to combine events. For each event an individual event memory entry is created, while reporting the data via UDS, the data is combined. **Tags:**atp.EnumerationLiteralIndex=1
eventCombination OnStorage	Event combination on storage is used to combine events. Only one memory entry exists for each DTC which is also reported via UDS. **Tags:**atp.EnumerationLiteralIndex=0

Table A.311: DiagnosticEventCombinationBehaviorEnum

Enumeration	DiagnosticEventCombinationReportingBehaviorEnum
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticCommonProps
Note	Select reporting format of events. Applicable only for Event Combination on Retrieval.
Aggregated by	DiagnosticCommonProps.eventCombinationReportingBehavior
Literal	Description
reportingIn ChronlogicalOrder OldestFirst	The reporting order for event combination on retrieval is the chronological storage order of the events **Tags:**atp.EnumerationLiteralIndex=0

Table A.312: DiagnosticEventCombinationReportingBehaviorEnum

Class	DiagnosticEventNeeds			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	Specifies the abstract needs on the configuration of the Diagnostic Event Manager for one diagnostic event. Its shortName can be regarded as a symbol identifying the diagnostic event from the viewpoint of the component or module which owns this element. In case the diagnostic event specifies a production error, the shortName shall be the name of the production error.			
Base	ARObject, DiagnosticCapabilityElement, Identifiable, MultilanguageReferrable, Referrable, Service Needs			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	Type	Mult.	Kind	Note
deferringFid	FunctionInhibitionNeeds	*	ref	This reference contains the link to a function identifier within the FiM which is used by the monitor before delivering a result.
diagEvent Debounce Algorithm	DiagEventDebounce Algorithm	0..1	aggr	Specifies the abstract need on the Debounce Algorithm applied by the Diagnostic Event Manager.
inhibitingFid	FunctionInhibitionNeeds	0..1	ref	This represents the primary Function Inhibition Identifier used for inhibition of the diagnostic monitor. The FID might either inhibit the monitoring of a symptom or the reporting of detected faults.
inhibiting SecondaryFid	FunctionInhibitionNeeds	*	ref	This represents the secondary Function Inhibition Identifier used for inhibition of the diagnostic monitor. Any of the FID inhibitions leads to an inhibition of the monitoring of a symptom or the reporting of detected faults.

▽

△

Class	DiagnosticEventNeeds			
prestored Freezeframe StoredInNvm	Boolean	0..1	attr	If the Event uses a prestored freeze-frame (using the operations PrestoreFreezeFrame and ClearPrestored FreezeFrame of the service interface DiagnosticMonitor) this attribute indicates if the Event requires the data to be stored in non-volatile memory. TRUE = Dem shall store the prestored data in non-volatile memory, FALSE = Data can be lost at shutdown (not stored in Nvm).
usesMonitor Data	Boolean	0..1	attr	This attribute defines whether additional monitor data shall be added to the reporting of events.

Table A.313: DiagnosticEventNeeds

Class	DiagnosticEventPortMapping			
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticMapping			
Note	Defines to which SWC service ports with DiagnosticEventInfoNeeds the DiagnosticEvent is mapped. **Tags:**atp.recommendedPackage=DiagnosticMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *DiagnosticSwMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
bswService Dependency	BswService DependencyIdent	0..1	ref	Reference to a BswServiceDependency that links Service Needs to BswModuleEntries.
diagnosticEvent	DiagnosticEvent	0..1	ref	Reference to the DiagnosticEvent that is assigned to SWC service ports with DiagnosticEventInfoNeeds.
swcFlatService Dependency	SwcService Dependency	0..1	ref	Reference to a SwcServiceDependencyType that links ServiceNeeds to SWC service ports.
swcService DependencyIn System	SwcService Dependency	0..1	iref	Instance reference to a SwcServiceDependency that links ServiceNeeds to SWC service ports. **InstanceRef implemented by:**SwcServiceDependency InSystemInstanceRef

Table A.314: DiagnosticEventPortMapping

Class	DiagnosticEventToDebounceAlgorithmMapping			
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticMapping			
Note	Defines which Debounce Algorithm is applicable for a DiagnosticEvent. **Tags:**atp.recommendedPackage=DiagnosticMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
debounce Algorithm	DiagnosticDebounce AlgorithmProps	0..1	ref	Reference to a DebounceAlgorithm assigned to a DiagnosticEvent.
diagnosticEvent	DiagnosticEvent	0..1	ref	Reference to a DiagnosticEvent to which a Debounce Algorithm is assigned.

Table A.315: DiagnosticEventToDebounceAlgorithmMapping

Class	DiagnosticEventToEnableConditionGroupMapping			
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticMapping			
Note	Defines which EnableConditionGroup is applicable for a DiagnosticEvent. **Tags:**atp.recommendedPackage=DiagnosticMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
diagnosticEvent	DiagnosticEvent	0..1	ref	Reference to a DiagnosticEvent to which an Enable ConditionGroup is assigned.
enableCondition Group	DiagnosticEnable ConditionGroup	0..1	ref	Reference to an EnableConditionGroup assigned to a DiagnosticEvent.

Table A.316: DiagnosticEventToEnableConditionGroupMapping

Class	DiagnosticEventToSecurityEventMapping			
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticMapping			
Note	This meta-class represents the ability to map a security event that is defined in the context of the Security Extract to a diagnostic event defined on the context of the DiagnosticExtract. **Tags:** atp.Status=candidate atp.recommendedPackage=DiagnosticMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
diagnosticEvent	DiagnosticEvent	0..1	ref	This reference identifies the applicable diagnostic event. **Tags:**atp.Status=candidate
securityEvent Props	SecurityEventContext Props	0..1	ref	This reference identifies the qualification of the applicable security event **Tags:**atp.Status=candidate

Table A.317: DiagnosticEventToSecurityEventMapping

Class	DiagnosticEventToStorageConditionGroupMapping			
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticMapping			
Note	Defines which StorageConditionGroup is applicable for a DiagnosticEvent. **Tags:**atp.recommendedPackage=DiagnosticMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
diagnosticEvent	DiagnosticEvent	0..1	ref	Reference to a DiagnosticEvent to which a Storage ConditionGroup is assigned.
storage ConditionGroup	DiagnosticStorage ConditionGroup	0..1	ref	Reference to a StorageConditionGroup assigned to a DiagnosticEvent.

Table A.318: DiagnosticEventToStorageConditionGroupMapping

Class	DiagnosticEventToTroubleCodeJ1939Mapping			
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticMapping::DiagnosticJ1939Mapping			
Note	By means of this meta-class it is possible to associate a DiagnosticEvent to a DiagnosticTroubleCode J1939. **Tags:**atp.recommendedPackage=DiagnosticMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
diagnosticEvent	DiagnosticEvent	0..1	ref	Reference to a DiagnosticEvent to which a J1939 Diagnostic Trouble Code is assigned.
troubleCode J1939	DiagnosticTroubleCode J1939	0..1	ref	Reference to a J1939 Diagnostic Trouble Code to which a DiagnosticEvent is assigned.

Table A.319: DiagnosticEventToTroubleCodeJ1939Mapping

Class	DiagnosticEventToTroubleCodeUdsMapping			
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticMapping			
Note	Defines which UDS Diagnostic Trouble Code is applicable for a DiagnosticEvent. **Tags:**atp.recommendedPackage=DiagnosticMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
diagnosticEvent	DiagnosticEvent	0..1	ref	Reference to a DiagnosticEvent to which a UDS Diagnostic Trouble Code is assigned.
troubleCodeUds	DiagnosticTroubleCode Uds	0..1	ref	Reference to an UDS Diagnostic Trouble Code assigned to a DiagnosticEvent.

Table A.320: DiagnosticEventToTroubleCodeUdsMapping

Class	DiagnosticExtendedDataRecord			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticExtendedDataRecord			
Note	Description of an extended data record. **Tags:**atp.recommendedPackage=DiagnosticExtendedDataRecords			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
customTrigger	String	0..1	attr	This attribute shall be taken to verbally describe the nature of the custom trigger.
recordElement	DiagnosticParameter	*	aggr	Defined DataElements in the extended record element.
recordNumber	PositiveInteger	0..1	attr	This attribute specifies an unique identifier for an extended data record.
trigger	DiagnosticRecord TriggerEnum	0..1	attr	This attribute specifies the primary trigger to allocate an event memory entry.

▽

△

Class	DiagnosticExtendedDataRecord			
update	Boolean	0..1	attr	This attribute defines when an extended data record is captured.
				True: This extended data record is captured every time.
				False: This extended data record is only captured for new event memory entries.

Table A.321: DiagnosticExtendedDataRecord

Class	DiagnosticFimAliasEventGroupMapping			
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticMapping::FimMapping			
Note	This meta-class represents the ability to map a DiagnosticFimEventGroup to a DiagnosticFimAliasEvent Group. By this means the "preliminary" modeling by way of a DiagnosticFimAliasEventGroup is further substantiated.			
	Tags:atp.recommendedPackage=DiagnosticFimAliasEventGroupMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
actualEvent	DiagnosticFimEvent Group	0..1	ref	This represents the reference to the actual summary event.
aliasEvent	DiagnosticFimAlias EventGroup	0..1	ref	This represents the reference to the alias summary event.

Table A.322: DiagnosticFimAliasEventGroupMapping

Class	DiagnosticFimAliasEventMapping			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticEvent			
Note	This meta-class represents the ability to model the mapping of a DiagnosticEvent to a DiagnosticAlias Event. By this means the "preliminary" modeling by way of a DiagnosticAliasEvent is further substantiated.			
	Tags:atp.recommendedPackage=DiagnosticFimEventMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
actualEvent	DiagnosticEvent	0..1	ref	This represents the reference to the actual diagnostic event.
aliasEvent	DiagnosticFimAlias Event	0..1	ref	This represents the reference to the alias event.

Table A.323: DiagnosticFimAliasEventMapping

Class	DiagnosticFimFunctionMapping
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticMapping::ServiceMapping
Note	This meta-class represents the ability to define a mapping between a function identifier (FID) and the corresponding SwcServiceDependency in the application software resp. basic software.
	Tags:atp.recommendedPackage=DiagnosticFimFunctionMappings

▽

△

Class	DiagnosticFimFunctionMapping			
Base	*ARElement*, ARObject, CollectableElement, *DiagnosticCommonElement*, DiagnosticMapping, DiagnosticSwMapping, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
mappedBsw Service Dependency	BswService DependencyIdent	0..1	ref	This is supposed to represent a reference to a Bsw ServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.
mappedFlatSwc Service Dependency	SwcService Dependency	0..1	ref	This represents the ability to refer to an AtomicSw ComponentType that is available without the definition of how it will be embedded into the component hierarchy.
mapped Function	DiagnosticFunction Identifier	0..1	ref	This represents the mapped FID.
mappedSwc Service Dependency	SwcService Dependency	0..1	iref	This represents the ability to point into the component hierarchy (under possible consideration of the root SoftwareComposition). **InstanceRef implemented by:**SwcServiceDependency InSystemInstanceRef

Table A.324: DiagnosticFimFunctionMapping

Class	DiagnosticFreezeFrame			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticFreezeFrame			
Note	This element describes combinations of DIDs for a non OBD relevant freeze frame. **Tags:**atp.recommendedPackage=DiagnosticFreezeFrames			
Base	*ARElement*, ARObject, CollectableElement, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
customTrigger	String	0..1	attr	This attribute shall be taken to verbally describe the nature of the custom trigger.
recordNumber	PositiveInteger	0..1	attr	This attribute defines a record number for a freeze frame record. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime
trigger	DiagnosticRecord TriggerEnum	0..1	attr	This attribute defines the primary trigger to allocate an event memory entry.
update	Boolean	0..1	attr	This attribute defines the approach when the freeze frame record is stored/updated. True: FreezeFrame record is captured every time. False: FreezeFrame record is only captured for new event memory entries.

Table A.325: DiagnosticFreezeFrame

Class	DiagnosticFunctionIdentifierInhibit			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Fim			
Note	This meta-class represents the ability to define the inhibition of a specific function identifier within the Fim configuration. **Tags:**atp.recommendedPackage=DiagnosticFunctionIdentifierInhibits			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
function Identifier	DiagnosticFunction Identifier	0..1	ref	This represents the corresponding function identifier.
inhibitionMask	DiagnosticInhibition MaskEnum	0..1	attr	This represents the value of the inhibition mask behavior.
inhibitSource	DiagnosticFunction InhibitSource	*	aggr	This represents a collection of DiagnosticFunctionInhibit Source that contribute to the configuration of the enclosing DiagnosticFunctionIdentiferInhibit.

Table A.326: DiagnosticFunctionIdentifierInhibit

Class	DiagnosticFunctionInhibitSource			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Fim			
Note	This meta-class represents the ability to define an inhibition source in the context of the Fim configuration.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DiagnosticFunctionIdentifierInhibit.inhibitSource			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
event	DiagnosticFimAlias Event	0..1	ref	This represents the alias event applicable for the referencing inhibition source.
eventGroup	DiagnosticFimAlias EventGroup	0..1	ref	This represents the event group applicable for the referencing inhibition source.

Table A.327: DiagnosticFunctionInhibitSource

Class	DiagnosticIOControl			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::IOControl			
Note	This represents an instance of the "I/O Control" diagnostic service. **Tags:**atp.recommendedPackage=DiagnosticIoControls			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
controlEnable MaskBit	DiagnosticControl EnableMaskBit	*	aggr	This aggregation represents the control mask record consisting of single bits.
dataIdentifier	DiagnosticDataIdentifier	0..1	ref	This represents the corresponding DiagnosticData Identifier
freezeCurrent State	Boolean	0..1	attr	Setting this attribute to true represents the ability of the Dcm to execute a freezeCurrentState.
ioControlClass	DiagnosticIoControl Class	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticIOControl in the given context.

$\bigtriangledown$

△

Class	DiagnosticIOControl			
resetToDefault	Boolean	0..1	attr	Setting this attribute to true represents the ability of the Dcm to execute a resetToDefault.
shortTerm Adjustment	Boolean	0..1	attr	Setting this attribute to true represents the ability of the Dcm to execute a shortTermAdjustment.

Table A.328: DiagnosticIOControl

Class	DiagnosticIndicator			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticIndicator			
Note	Definition of an indicator. **Tags:**atp.recommendedPackage=DiagnosticIndicators			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
type	DiagnosticIndicatorType Enum	0..1	attr	Defines the type of the indicator. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime

Table A.329: DiagnosticIndicator

Class	DiagnosticInfoType			
Package	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
Note	This meta-class represents the ability to model an OBD info type. **Tags:**atp.recommendedPackage=DiagnosticInfoTypes			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
dataElement	DiagnosticParameter	*	aggr	This represents the data associated with the enclosing DiagnosticInfoType. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=dataElement.bitOffset, data Element.ident.shortName
id	PositiveInteger	0..1	attr	This attribute represents the value of InfoType (see SAE J1979-DA).

Table A.330: DiagnosticInfoType

Class	DiagnosticIoControlNeeds			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	Specifies the general needs on the configuration of the Diagnostic Communication Manager (DCM) which are not related to a particular item (e.g. a PID). The main use case is the mapping of service ports to the Dcm which are not related to a particular item.			
Base	*ARObject*, *DiagnosticCapabilityElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *Service Needs*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	Type	Mult.	Kind	Note

▽

△

Class	DiagnosticIoControlNeeds			
currentValue	DiagnosticValueNeeds	0..1	ref	Reference to the DiagnosticValueNeeds indicating the access to the current value via signalBasedDiagnostics.
freezeCurrent StateSupported	Boolean	0..1	attr	This attribute determines, if the referenced port supports temporary freezing of I/O value.
resetToDefault Supported	Boolean	0..1	attr	This represents a flag for the existence of the ResetTo Default operation in the service interface.
shortTerm Adjustment Supported	Boolean	0..1	attr	This attribute determines, if the referenced port supports temporarily setting of I/O value to a specific value provided by the diagnostic tester.

Table A.331: DiagnosticIoControlNeeds

Class	DiagnosticIumprGroup			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticEvent			
Note	This meta-class represents the ability to model a IUMPR groups. **Tags:**atp.recommendedPackage=DiagnosticIumprGroups			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
iumpr	DiagnosticIumpr	*	ref	This reference collects DiagnosticIumpr to a Diagnostic IumprGroup.
iumprGroup Identifier	DiagnosticIumprGroup Identifier	0..1	aggr	This aggregation allows for the variant modeling of the groupIdentifier. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=iumprGroupIdentifier.groupId, iumprGroup Identifier.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.332: DiagnosticIumprGroup

Class	DiagnosticIumprGroupIdentifier			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticEvent			
Note	This meta-class provides the ability to the define the group identifier for an IumprGroup.			
Base	*ARObject*			
Aggregated by	DiagnosticIumprGroup.iumprGroupIdentifier			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
groupId	NameToken	0..1	attr	This attribute shall be taken to define an identifier for the IUMPR group. Please note that the value of this identifier is driven by regulations outside the scope of AUTOSAR and can therefore not be limited to the set of characters suitable for a shortName. **Stereotypes:** atpIdentityContributor

Table A.333: DiagnosticIumprGroupIdentifier

Class	DiagnosticIumprToFunctionIdentifierMapping			
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticMapping			
Note	This meta-class represents the ability to associate a DiagnosticFunctionIdentifier with a DiagnosticIumpr. **Tags:**atp.recommendedPackage=DiagnosticMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
function Identifier	DiagnosticFunction Identifier	0..1	ref	This reference identifies the applicable Diagnostic FunctionIdentifier.
iumpr	DiagnosticIumpr	0..1	ref	This reference identifies the applicable DiagnosticIumpr.

Table A.334: DiagnosticIumprToFunctionIdentifierMapping

Class	DiagnosticJ1939Node			
Package	M2::AUTOSARTemplates::DiagnosticExtract::J1939			
Note	This meta-class represents the diagnostic configuration of a J1939 Nm node, which in turn represents a "virtual Ecu" on the J1939 communication bus. **Tags:**atp.recommendedPackage=DiagnosticJ1939Nodes			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
nmNode	J1939NmNode	0..1	ref	This represents the reference to the "virtual Ecu" to which the enclosing DiagnosticJ1939Node is associated.

Table A.335: DiagnosticJ1939Node

Class	DiagnosticJ1939Spn			
Package	M2::AUTOSARTemplates::DiagnosticExtract::J1939			
Note	This meta-class represents the ability to model a J1939 Suspect Parameter Number (SPN). **Tags:**atp.recommendedPackage=DiagnosticJ1939Spns			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
spn	PositiveInteger	0..1	attr	This attribute represents the concrete numerical identification for the enclosing SPN.

Table A.336: DiagnosticJ1939Spn

Class	DiagnosticMasterToSlaveEventMapping			
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticMapping			
Note	This meta-class provides the ability to map a master diagnostic event with a slave diagnostic event such that reporting of the master event with a given value also reports the slave event with the same value **Tags:**atp.recommendedPackage=DiagnosticMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			

▽

△

Class	DiagnosticMasterToSlaveEventMapping			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
masterEvent	DiagnosticEvent	0..1	ref	This represents the master diagnostic event.
slaveEvent	DiagnosticEvent	0..1	ref	This represents the slave diagnostic event.

Table A.337: DiagnosticMasterToSlaveEventMapping

Class	*DiagnosticMemoryAddressableRangeAccess* (abstract)			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress			
Note	This abstract base class			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMemoryByAddress*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	DiagnosticReadMemoryByAddress, DiagnosticRequestDownload, DiagnosticRequestUpload, Diagnostic WriteMemoryByAddress			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
memoryRange	DiagnosticMemory Identifier	*	ref	This represents the formal description of the memory segment to which the DiagnosticMemoryByAddress applies.

Table A.338: DiagnosticMemoryAddressableRangeAccess

Class	*DiagnosticMemoryDestination* (abstract)			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticMemoryDestination			
Note	This abstract meta-class represents a possible memory destination for a diagnostic event.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Subclasses	DiagnosticMemoryDestinationPrimary, DiagnosticMemoryDestinationUserDefined			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
agingRequires TestedCycle	Boolean	0..1	attr	Defines whether the aging cycle counter is processed every aging cycles or else only tested aging cycle are considered. If the attribute is set to TRUE: only tested aging cycle are considered for aging cycle counter. If the attribute is set to FALSE: aging cycle counter is processed every aging cycle.
clearDtc Limitation	DiagnosticClearDtc LimitationEnum	0..1	attr	Defines the scope of the DEM_ClearDTC Api.
dtcStatus AvailabilityMask	PositiveInteger	0..1	attr	Mask for the supported DTC status bits by the Dem.
event Displacement Strategy	DiagnosticEvent DisplacementStrategy Enum	0..1	attr	This attribute defines, whether support for event displacement is enabled or not, and which displacement strategy is followed.
maxNumberOf EventEntries	PositiveInteger	0..1	attr	This attribute fixes the maximum number of event entries in the fault memory.
memoryEntry StorageTrigger	DiagnosticMemoryEntry StorageTriggerEnum	0..1	attr	Describes the trigger to allocate an event memory entry.

▽

△

Class	DiagnosticMemoryDestination (abstract)			
statusBit HandlingTest FailedSinceLast Clear	DiagnosticStatusBit HandlingTestFailed SinceLastClearEnum	0..1	attr	This attribute defines, whether the aging and displacement mechanism shall be applied to the "Test FailedSinceLastClear" status bits.
statusBit StorageTest Failed	Boolean	0..1	attr	This parameter is used to activate/deactivate the permanent storage of the "TestFailed" status bits. true: storage activated false: storage deactivated
typeOfFreeze FrameRecord Numeration	DiagnosticTypeOf FreezeFrameRecord NumerationEnum	0..1	attr	This attribute defines the type of assigning freeze frame record numbers for event-specific freeze frame records.

Table A.339: DiagnosticMemoryDestination

Class	DiagnosticMemoryDestinationPrimary			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticMemoryDestination			
Note	This represents a primary memory for a diagnostic event. **Tags:**atp.recommendedPackage=DiagnosticMemoryDestinations			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMemory Destination*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
typeOfDtc Supported	DiagnosticTypeOfDtc SupportedEnum	0..1	attr	This attribute defines the format returned by Dem_Dcm GetTranslationType and does not relate to/influence the supported Dem functionality.

Table A.340: DiagnosticMemoryDestinationPrimary

Class	DiagnosticMemoryDestinationUserDefined			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticMemoryDestination			
Note	This represents a user-defined memory for a diagnostic event. **Tags:**atp.recommendedPackage=DiagnosticMemoryDestinations			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMemory Destination*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
authentication Role	DiagnosticAuthRole	0..1	ref	This reference identifies the authenticationRole applicable for the enclosing DiagnosticMemoryDestinationUser Defined.
memoryId	PositiveInteger	0..1	attr	This represents the identifier of the user-defined memory.

Table A.341: DiagnosticMemoryDestinationUserDefined

Class	DiagnosticMemoryIdentifier
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress
Note	This meta-class represents the ability to define memory properties from the diagnostics point of view. **Tags:**atp.recommendedPackage=DiagnosticMemoryByAdresss

▽

△

Class	DiagnosticMemoryIdentifier			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
access Permission	DiagnosticAccess Permission	0..1	ref	This represents that access permission defined for the specific DiagnosticMemoryIdentifier.
id	PositiveInteger	0..1	attr	This represents the identification of the memory segment.
memoryHigh Address	PositiveInteger	0..1	attr	This represents the upper bound for addresses of the memory segment.
memoryHigh AddressLabel	String	0..1	attr	This represents a symbolic label for the upper bound for addresses of the memory segment.
memoryLow Address	PositiveInteger	0..1	attr	This represents the lower bound for addresses of the memory segment.
memoryLow AddressLabel	String	0..1	attr	This represents a symbolic label for the lower bound for addresses of the memory segment.

Table A.342: DiagnosticMemoryIdentifier

Class	DiagnosticOperationCycle			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticOperationCycle			
Note	Definition of an operation cycle that is the base of the event qualifying and for Dem scheduling. **Tags:**atp.recommendedPackage=DiagnosticOperationCycles			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
type	DiagnosticOperation CycleTypeEnum	0..1	attr	Operation cycles types for the Dem.

Table A.343: DiagnosticOperationCycle

Class	DiagnosticOperationCyclePortMapping			
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticMapping			
Note	Defines to which SWC service ports with DiagnosticOperationCycleNeeds the DiagnosticOperationCycle is mapped. **Tags:**atp.recommendedPackage=DiagnosticMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *DiagnosticSwMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
operationCycle	DiagnosticOperation Cycle	0..1	ref	Reference to the DiagnosticOperationCycle that is assigned to SWC service ports with DiagnosticOperation CycleNeeds.
swcFlatService Dependency	SwcService Dependency	0..1	ref	Reference to a SwcServiceDependencyType that links ServiceNeeds to SWC service ports.

▽

△

Class	DiagnosticOperationCyclePortMapping			
swcService DependencyIn System	SwcService Dependency	0..1	iref	Instance reference to a SwcServiceDependency that links ServiceNeeds to SWC service ports. **InstanceRef implemented by:**SwcServiceDependency InSystemInstanceRef

Table A.344: DiagnosticOperationCyclePortMapping

Class	DiagnosticParameter			
Package	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
Note	This meta-class represents the ability to describe information relevant for the execution of a specific diagnostic service, i.e. it can be taken to parameterize the service.			
Base	ARObject, *DiagnosticAbstractParameter*			
Aggregated by	DiagnosticDataIdentifier.dataElement, DiagnosticExtendedDataRecord.recordElement, DiagnosticInfo Type.dataElement, DiagnosticParameterIdentifier.dataElement, DiagnosticRequestRoutineResults. request, DiagnosticRequestRoutineResults.response, DiagnosticStartRoutine.request, DiagnosticStart Routine.response, DiagnosticStopRoutine.request, DiagnosticStopRoutine.response			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
ident	DiagnosticParameter Ident	0..1	aggr	The aggregation in the role ident provides the ability to make the DiagnosticAbstractParameter identifiable. From the semantical point of view, the AbstractDiagnostic Parameter is considered a first-class Identifiable and therefore the aggregation in the role ident shall always exist (until it may be possible to let AbstractDiagnostic Parameter directly inherit from Identifiable). **Stereotypes:** atpIdentityContributor
supportInfo	DiagnosticParameter SupportInfo	0..1	aggr	This attribute represents the ability to define which bit of the support info byte is representing this part of the PID.

Table A.345: DiagnosticParameter

Class	DiagnosticParameterElement			
Package	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
Note	This meta-class represents an element of a DiagnosticParameter if the DiagnosticParameter represents a structure.			
Base	ARObject, *DiagnosticAbstractParameter*, *DiagnosticServiceMappingDiagTarget*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DiagnosticParameterElement.subElement, DiagnosticParameterIdent.subElement			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
arraySize	PositiveInteger	0..1	attr	This attribute indicates that the enclosing Diagnostic ParameterElement represents an array and configures the array size in terms of the number of elements of the array.
subElement	DiagnosticParameter Element	*	aggr	This collection represents the sub-elements on the next lower level.

Table A.346: DiagnosticParameterElement

Class	DiagnosticParameterElementAccess			
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticMapping::ServiceMapping			
Note	This meta-class acts as a single point for defining structured references to a specific Diagnostic ParameterElement.			
Base	*ARObject*			
Aggregated by	DiagnosticServiceDataMapping.parameterElementAccess, DiagnosticServiceSwMapping.parameter ElementAccess			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
contextElement (ordered)	DiagnosticParameter Element	*	ref	This represents the context of an applicable payload that corresponds to the referenced DataPrototype in the role mappedDataElement. **Tags:**xml.sequenceOffset=10
targetElement	DiagnosticParameter Element	0..1	ref	This represents the target reference of an applicable payload that corresponds to the referenced Data Prototype in the role mappedDataElement. **Tags:**xml.sequenceOffset=20

Table A.347: DiagnosticParameterElementAccess

Class	DiagnosticParameterIdent			
Package	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
Note	This meta-class has been created to introduce the ability to become referenced into the meta-class AbstractDiagnosticParameter without breaking backwards compatibility.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *DiagnosticServiceMappingDiagTarget*, *Ident Caption*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, DiagnosticParameter.ident			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
subElement	DiagnosticParameter Element	*	aggr	This collection represents the subElements on the top level.

Table A.348: DiagnosticParameterIdent

Class	DiagnosticParameterIdentifier			
Package	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
Note	This meta-class represents the ability to model a diagnostic parameter identifier (PID) for the purpose of executing on-board diagnostics (OBD). **Tags:**atp.recommendedPackage=DiagnosticParameterIdentifiers			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dataElement	DiagnosticParameter	*	aggr	This represents the data carried by the Diagnostic ParameterIdentifier. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=dataElement.bitOffset, data Element.ident.shortName, dataElement.variation Point.shortLabel vh.latestBindingTime=postBuild
id	PositiveInteger	0..1	attr	This is the numerical identifier used to identify the DiagnosticParameterIdentifier in the scope of diagnostic workflow (see SAE J1979-DA).

▽

Document ID 635: AUTOSAR_TR_ClassicAutosarModelConstraints

△

Class	DiagnosticParameterIdentifier			
pidSize	PositiveInteger	0..1	attr	The size of the entire PID can be greater than the sum of the data elements because padding might be applied. Unit: byte.
supportInfoByte	DiagnosticSupportInfoByte	0..1	aggr	This represents the supported information associated with the DiagnosticParameterIdentifier.

Table A.349: DiagnosticParameterIdentifier

Class	DiagnosticPeriodicRate			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ReadDataByPeriodicID			
Note	This represents the ability to define a periodic rate for the specification of the "read data by periodic ID" diagnostic service.			
Base	ARObject			
Aggregated by	DiagnosticReadDataByPeriodicIDClass.periodicRate			
Attribute	Type	Mult.	Kind	Note
period	TimeValue	0..1	attr	This represents the period of the DiagnosticPeriodicRate in seconds.
periodicRateCategory	DiagnosticPeriodicRateCategoryEnum	0..1	attr	This attribute represents the category of the periodic rate.

Table A.350: DiagnosticPeriodicRate

Class	DiagnosticProofOfOwnership			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::Authentication			
Note	This meta-class represents the subfunction to provide proof of ownership. Tags:atp.recommendedPackage=DiagnosticAuthentications			
Base	ARElement, ARObject, CollectableElement, DiagnosticAuthentication, DiagnosticCommonElement, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.351: DiagnosticProofOfOwnership

Class	DiagnosticProtocol			
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticContribution			
Note	This meta-class represents the ability to define a diagnostic protocol. Tags:atp.recommendedPackage=DiagnosticProtocols			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note

▽

Document ID 635: AUTOSAR_TR_ClassicAutosarModelConstraints

△

Class	DiagnosticProtocol			
diagnostic Connection	DiagnosticConnection	*	ref	This represents the collection of applicable Diagnostic Connections for this DiagnosticProtocol. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=diagnosticConnection.diagnosticConnection, diagnosticConnection.variationPoint.shortLabel vh.latestBindingTime=postBuild
priority	PositiveInteger	0..1	attr	This represents the priority of the diagnostic protocol in comparison to other diagnostic protocols. Lower numeric values represent higher protocol priority: • 0 - Highest protocol priority • 255 - Lowest protocol priority **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime
protocolKind	NameToken	0..1	attr	This identifies the applicable protocol.
sendRespPend OnTransToBoot	Boolean	0..1	attr	The purpose of this attribute is to define whether or not the ECU should send a NRC 0x78 (response pending) before transitioning to the bootloader (in this case the attribute shall be set to "true") or if the transition shall be initiated without sending NRC 0x78 (in this case the attribute shall be set to "false"). **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime
serviceTable	DiagnosticServiceTable	0..1	ref	This represents the service table applicable for the given diagnostic protocol. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=serviceTable.diagnosticServiceTable, serviceTable.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.352: DiagnosticProtocol

Class	DiagnosticReadDataByIdentifier			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::DataByIdentifier			
Note	This represents an instance of the "Read Data by Identifier" diagnostic service. **Tags:**atp.recommendedPackage=DiagnosticDataByIdentifiers			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticDataByIdentifier*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
readClass	DiagnosticReadDataBy IdentifierClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticReadDataBy Identifier in the given context.

Table A.353: DiagnosticReadDataByIdentifier

Class	DiagnosticReadDataByPeriodicIDClass			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ReadDataByPeriodicID			
Note	This meta-class contains attributes shared by all instances of the "Read Data by periodic Identifier" diagnostic service. **Tags:**atp.recommendedPackage=DiagnosticReadDataByPeriodicIds			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
maxPeriodicDidToRead	PositiveInteger	0..1	attr	This represents the maximum number of data identifiers that can be included in one request.
periodicRate	DiagnosticPeriodicRate	*	aggr	This represents the description of a collection of periodic rates in which the service can be executed.
schedulerMaxNumber	PositiveInteger	0..1	attr	This represents the maximum number of periodic data identifiers that can be scheduled in parallel.

Table A.354: DiagnosticReadDataByPeriodicIDClass

Class	DiagnosticReadScalingDataByIdentifier			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::DataByIdentifier			
Note	This represents an instance of the "Read Scaling Data by Identifier" diagnostic service. **Tags:**atp.recommendedPackage=DiagnosticDataByIdentifiers			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticDataByIdentifier*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
readScalingDataClass	DiagnosticReadScalingDataByIdentifierClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticReadScalingDataByIdentifier in the given context.

Table A.355: DiagnosticReadScalingDataByIdentifier

Enumeration	DiagnosticRecordTriggerEnum
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticFreezeFrame
Note	Triggers to allocate an event memory entry.
Aggregated by	DiagnosticExtendedDataRecord.trigger, DiagnosticFreezeFrame.trigger
Literal	**Description**
confirmed	capture on "Confirmed" **Tags:**atp.EnumerationLiteralIndex=0
custom	implement custom capture **Tags:**atp.EnumerationLiteralIndex=4
fdcThreshold	capture on "FDC Threshold" **Tags:**atp.EnumerationLiteralIndex=1
pending	capture on "Pending" **Tags:**atp.EnumerationLiteralIndex=2

$\bigtriangledown$

△

Enumeration	DiagnosticRecordTriggerEnum
testFailed	capture on "Test Failed" **Tags:**atp.EnumerationLiteralIndex=3
testFailedThis OperationCycle	Test Failed This Operation Cycle. **Tags:**atp.EnumerationLiteralIndex=5
testPassed	Capture on testFailed bit transition 1 –> 0. **Tags:**atp.EnumerationLiteralIndex=6

Table A.356: DiagnosticRecordTriggerEnum

Class	DiagnosticRequestControlOfOnBoardDevice			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x08_RequestControlOfOnBoard Device			
Note	This meta-class represents the ability to model an instance of the OBD mode 0x08 service. **Tags:**atp.recommendedPackage=DiagnosticRequestControlOfOnBoardDevices			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
requestControl OfOnBoard DeviceClass	DiagnosticRequest ControlOfOnBoard DeviceClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequestControlOf OnBoardDevice in the given context.
testId	DiagnosticTestRoutine Identifier	0..1	ref	This represents the test Id for the mode 0x08.

Table A.357: DiagnosticRequestControlOfOnBoardDevice

Class	DiagnosticRequestCurrentPowertrainData			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x01_RequestCurrentPowertrain DiagnosticData			
Note	This meta-class represents the ability to model an instance of the OBD mode 0x01 service. **Tags:**atp.recommendedPackage=DiagnosticRequestCurrentPowertrainDatas			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
pid	DiagnosticParameter Identifier	0..1	ref	This represents the PID associated with this instance of the OBD mode 0x01 service.
requestCurrent Powertrain DiagnosticData Class	DiagnosticRequest CurrentPowertrainData Class	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequestCurrent PowertrainData in the given context.

Table A.358: DiagnosticRequestCurrentPowertrainData

Class	DiagnosticRequestDownload
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress
Note	This represents an instance of the "Request Download" diagnostic service. **Tags:**atp.recommendedPackage=DiagnosticMemoryByAdresss
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMemory AddressableRangeAccess*, *DiagnosticMemoryByAddress*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*
Aggregated by	ARPackage.element

Attribute	Type	Mult.	Kind	Note
request DownloadClass	DiagnosticRequest DownloadClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequestDownload in the given context.

Table A.359: DiagnosticRequestDownload

Class	DiagnosticRequestEmissionRelatedDTC
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x03_0x07_RequestEmission RelatedDTC
Note	This meta-class represents the ability to model an instance of the OBD mode 0x03/0x07 service. **Tags:**atp.recommendedPackage=DiagnosticRequestEmissionRelatedDTCs
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*
Aggregated by	ARPackage.element

Attribute	Type	Mult.	Kind	Note
request Emission RelatedDtc Class	DiagnosticRequest EmissionRelatedDTC Class	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequestEmission RelatedDTC in the given context.

Table A.360: DiagnosticRequestEmissionRelatedDTC

Class	DiagnosticRequestEmissionRelatedDTCPermanentStatus
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x0A_RequestEmissionRelated DTCPermanentStatus
Note	This meta-class represents the ability to model an instance of the OBD mode 0x0A service. **Tags:**atp.recommendedPackage=DiagnosticRequestEmissionRelatedDTCPermanentStatuss
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*
Aggregated by	ARPackage.element

Attribute	Type	Mult.	Kind	Note
request Emission RelatedDtc Class Permanent Status	DiagnosticRequest EmissionRelatedDTC PermanentStatusClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequestEmission RelatedDTCPermanentStatus in the given context.

Table A.361: DiagnosticRequestEmissionRelatedDTCPermanentStatus

Class	DiagnosticRequestOnBoardMonitoringTestResults			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x06_RequestOnBoard MonitoringTestResults			
Note	This meta-class represents the ability to model an instance of the OBD mode 0x06 service. **Tags:**atp.recommendedPackage=DiagnosticRequestOnBoardMonitoringTestResultss			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
diagnosticTest Result	DiagnosticTestResult	*	ref	This reference identifies the applicable collection of test identifiers for setting up a request message for mode 0x06.
requestOn Board MonitoringTest ResultsClass	DiagnosticRequestOn BoardMonitoringTest ResultsClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequestOnBoard MonitoringTestResults in the given context.

Table A.362: DiagnosticRequestOnBoardMonitoringTestResults

Class	DiagnosticRequestPowertrainFreezeFrameData			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x02_RequestPowertrainFreeze FrameData			
Note	This meta-class represents the ability to model an instance of the OBD mode 0x02 service. **Tags:**atp.recommendedPackage=DiagnosticPowertrainFreezeFrames			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
freezeFrame	DiagnosticPowertrain FreezeFrame	0..1	ref	This represents the associated freeze-frame.
request Powertrain FreezeFrame Data	DiagnosticRequest PowertrainFreezeFrame DataClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequestPowertrain FreezeFrameData in the given context.

Table A.363: DiagnosticRequestPowertrainFreezeFrameData

Class	DiagnosticRequestRoutineResults			
Package	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
Note	This meta-class represents the ability to define the result of a diagnostic routine execution.			
Base	*ARObject*, *DiagnosticRoutineSubfunction*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DiagnosticRoutine.requestResult			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
request	DiagnosticParameter	*	aggr	This represents the request parameters.
response	DiagnosticParameter	*	aggr	This represents the response parameters.

Table A.364: DiagnosticRequestRoutineResults

Class	DiagnosticRequestUpload			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress			
Note	This represents an instance of the "Request Upload" diagnostic service. **Tags:**atp.recommendedPackage=DiagnosticMemoryByAdresss			
Base	*ARElement*, ARObject, CollectableElement, *DiagnosticCommonElement*, *DiagnosticMemory AddressableRangeAccess*, DiagnosticMemoryByAddress, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
requestUpload Class	DiagnosticRequest UploadClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequestUpload in the given context.

Table A.365: DiagnosticRequestUpload

Class	DiagnosticRequestVehicleInfo			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x09_RequestVehicleInformation			
Note	This meta-class represents the ability to model an instance of the OBD mode 0x09 service. **Tags:**atp.recommendedPackage=DiagnosticRequestVehicleInfos			
Base	*ARElement*, ARObject, CollectableElement, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
infoType	DiagnosticInfoType	0..1	ref	This represents the info type associated with the mode 0x09 service.
requestVehicle Information Class	DiagnosticRequest VehicleInfoClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequesVehicleInfo in the given context.

Table A.366: DiagnosticRequestVehicleInfo

Class	DiagnosticRoutine			
Package	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
Note	This meta-class represents the ability to define a diagnostic routine. **Tags:**atp.recommendedPackage=DiagnosticRoutines			
Base	*ARElement*, ARObject, CollectableElement, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
id	PositiveInteger	0..1	attr	This is the numerical identifier used to identify the DiagnosticRoutine in the scope of diagnostic workflow **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime
requestResult	DiagnosticRequest RoutineResults	0..1	aggr	This represents the ability to request the result of a running routine.

▽

△

Class	DiagnosticRoutine			
routineInfo	PositiveInteger	0..1	attr	This represents the routine info byte. The info byte contains a manufacturer-specific value (for the identification of record identifiers) that is reported to the tester. Other use cases for this attribute are mentioned in ISO 27145 and ISO 26021.
start	DiagnosticStartRoutine	0..1	aggr	This represents the ability to start a routine
stop	DiagnosticStopRoutine	0..1	aggr	This represents the ability to stop a running routine.

Table A.367: DiagnosticRoutine

Class	DiagnosticRoutineControl			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::RoutineControl			
Note	This represents an instance of the "Routine Control" diagnostic service. **Tags:**atp.recommendedPackage=DiagnosticRoutineControls			
Base	*ARElement*, ARObject, CollectableElement, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
routine	DiagnosticRoutine	0..1	ref	This refers to the applicable DiagnosticRoutine.
routineControl Class	DiagnosticRoutine ControlClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRoutineControl in the given context.

Table A.368: DiagnosticRoutineControl

Class	DiagnosticRoutineNeeds			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	Specifies the general needs on the configuration of the Diagnostic Communication Manager (Dcm) which are not related to a particular item (e.g. a PID). The main use case is the mapping of service ports to the Dcm which are not related to a particular item.			
Base	ARObject, DiagnosticCapabilityElement, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *Service Needs*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	Type	Mult.	Kind	Note
diagRoutine Type	DiagnosticRoutineType Enum	0..1	attr	This denotes the type of diagnostic routine which is implemented by the referenced server port.

Table A.369: DiagnosticRoutineNeeds

Enumeration	DiagnosticRoutineTypeEnum
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds
Note	This enumerator specifies the different types of diagnostic routines.
Aggregated by	DiagnosticRoutineNeeds.diagRoutineType
Literal	Description

▽

△

Enumeration	DiagnosticRoutineTypeEnum
asynchronous	This indicates that the diagnostic server is not blocked while the diagnostic routine is running. **Tags:**atp.EnumerationLiteralIndex=0
synchronous	This indicates that the diagnostic routine blocks the diagnostic server in the ECU while the routine is running. **Tags:**atp.EnumerationLiteralIndex=1

Table A.370: DiagnosticRoutineTypeEnum

Class	DiagnosticSecurityAccess			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::SecurityAccess			
Note	This represents an instance of the "Security Access" diagnostic service. **Tags:**atp.recommendedPackage=DiagnosticSecurityAccesss			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
requestSeedId	PositiveInteger	0..1	attr	This would be 0x01, 0x03, 0x05, ... The sendKey id can be computed by adding 1 to the requestSeedId
securityAccess Class	DiagnosticSecurity AccessClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticSecurityAccess in the given context.
securityDelay TimeOnBoot	TimeValue	0..1	attr	Start delay timer on power on in seconds. This delay indicates the time at ECU boot power-on time where the Dcm remains in the default session and does not accept a security access.
securityLevel	DiagnosticSecurityLevel	0..1	ref	This reference identifies the applicable security level for the security access. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=securityLevel

Table A.371: DiagnosticSecurityAccess

Class	DiagnosticSecurityAccessClass			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::SecurityAccess			
Note	This meta-class contains attributes shared by all instances of the "Security Access" diagnostic service. **Tags:**atp.recommendedPackage=DiagnosticSecurityAccesss			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceClass*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.372: DiagnosticSecurityAccessClass

Class	DiagnosticSecurityEventReportingModeMapping			
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticMapping::ServiceMapping			
Note	This meta-class represents the ability to associate a location in a DID with a security event. The purpose of this mapping is that the location in the DID contains the setting of the reporting mode for the specific security event. This means that the reporting mode of the security event can be set via the diagnostic service WriteDataByIdentifier. **Tags:** atp.Status=candidate atp.recommendedPackage=DiagnosticMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dataElement	DiagnosticDataElement	0..1	ref	This reference identifies the data element that carries the information about the reporting mode. **Tags:**atp.Status=candidate
securityEvent	SecurityEventContext Props	0..1	ref	This reference identifies the mapped security event. **Tags:**atp.Status=candidate

Table A.373: DiagnosticSecurityEventReportingModeMapping

Class	DiagnosticSecurityLevel			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm			
Note	This meta-class represents the ability to define a security level considered for diagnostic purposes. **Tags:**atp.recommendedPackage=DiagnosticSecurityLevels			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
accessData RecordSize	PositiveInteger	0..1	attr	This represents the size of the AccessDataRecord used in GetSeed. Unit:byte.
keySize	PositiveInteger	0..1	attr	This represents the size of the security key. Unit: byte.
numFailed SecurityAccess	PositiveInteger	0..1	attr	This represents the number of failed security accesses after which the delay time is activated.
securityDelay Time	TimeValue	0..1	attr	This represents the delay time after a failed security access. Unit: second.
seedSize	PositiveInteger	0..1	attr	This represents the size of the security seed. Unit: byte.

Table A.374: DiagnosticSecurityLevel

Class	*DiagnosticServiceClass* (abstract)
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::CommonService
Note	This meta-class provides the ability to define common properties that are shared among all instances of sub-classes of DiagnosticServiceInstance.
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*

▽

△

Class	DiagnosticServiceClass (abstract)
Subclasses	DiagnosticAuthenticationClass, DiagnosticClearDiagnosticInformationClass, DiagnosticClearReset EmissionRelatedInfoClass, DiagnosticComControlClass, DiagnosticControlDTCSettingClass, Diagnostic CustomServiceClass, DiagnosticDataTransferClass, DiagnosticDynamicallyDefineDataIdentifierClass, DiagnosticEcuResetClass, DiagnosticIoControlClass, DiagnosticReadDTCInformationClass, Diagnostic ReadDataByIdentifierClass, DiagnosticReadDataByPeriodicIDClass, DiagnosticReadMemoryByAddress Class, DiagnosticReadScalingDataByIdentifierClass, DiagnosticRequestControlOfOnBoardDeviceClass, DiagnosticRequestCurrentPowertrainDataClass, DiagnosticRequestDownloadClass, DiagnosticRequest EmissionRelatedDTCClass, DiagnosticRequestEmissionRelatedDTCPermanentStatusClass, Diagnostic RequestFileTransferClass, DiagnosticRequestOnBoardMonitoringTestResultsClass, DiagnosticRequest PowertrainFreezeFrameDataClass, DiagnosticRequestUploadClass, DiagnosticRequestVehicleInfo Class, DiagnosticResponseOnEventClass, DiagnosticRoutineControlClass, DiagnosticSecurityAccess Class, DiagnosticSessionControlClass, DiagnosticTransferExitClass, DiagnosticWriteDataByIdentifier Class, DiagnosticWriteMemoryByAddressClass
Aggregated by	ARPackage.element

Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.375: DiagnosticServiceClass

Class	DiagnosticServiceDataMapping
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticMapping::ServiceMapping
Note	This represents the ability to define a mapping of a diagnostic service to a software-component. This kind of service mapping is applicable for the usage of SenderReceiverInterfaces or event/notifier semantics in ServiceInterfaces on the adaptive platform. **Tags:**atp.recommendedPackage=DiagnosticServiceMappings
Base	*ARElement*, ARObject, CollectableElement, *DiagnosticCommonElement*, *DiagnosticMapping*, *DiagnosticSwMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*
Aggregated by	ARPackage.element

Attribute	Type	Mult.	Kind	Note
diagnosticData Element	DiagnosticDataElement	0..1	ref	This represents the applicable payload that corresponds to the referenced DataPrototype in the role mappedData Element or (in case of a usage on the adaptive platform) mappedApDataElement.
diagnostic Parameter	DiagnosticParameter Ident	0..1	ref	This represents the applicable payload that corresponds to the referenced DataPrototype in the role mappedData Element. **Tags:**xml.sequenceOffset=20
mappedData Element	DataPrototype	0..1	iref	This represents the dataElement in the application software that is accessed for diagnostic purpose. This role is applicable on the classic platform. **InstanceRef implemented by:**DataPrototypeInSystem InstanceRef
parameter ElementAccess	DiagnosticParameter ElementAccess	0..1	aggr	This aggregation represents the single point of access to the reference to one specific DiagnosticParameter Element.

Table A.376: DiagnosticServiceDataMapping

Class	DiagnosticServiceInstance (abstract)			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::CommonService			
Note	This represents a concrete instance of a diagnostic service.			
Base	*ARElement*, ARObject, CollectableElement, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Subclasses	*DiagnosticAuthentication*, DiagnosticClearDiagnosticInformation, DiagnosticClearResetEmissionRelated Info, DiagnosticComControl, DiagnosticControlDTCSetting, DiagnosticCustomServiceInstance, *DiagnosticDataByIdentifier*, DiagnosticDynamicallyDefineDataIdentifier, DiagnosticEcuReset, Diagnostic IOControl, *DiagnosticMemoryByAddress*, DiagnosticReadDTCInformation, DiagnosticReadDataBy PeriodicID, DiagnosticRequestControlOfOnBoardDevice, DiagnosticRequestCurrentPowertrainData, DiagnosticRequestEmissionRelatedDTC, DiagnosticRequestEmissionRelatedDTCPermanentStatus, DiagnosticRequestFileTransfer, DiagnosticRequestOnBoardMonitoringTestResults, DiagnosticRequest PowertrainFreezeFrameData, DiagnosticRequestVehicleInfo, DiagnosticResponseOnEvent, Diagnostic RoutineControl, DiagnosticSecurityAccess, DiagnosticSessionControl			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
access Permission	DiagnosticAccess Permission	0..1	ref	This represents the collection of DiagnosticAccess Permissions that allow for the execution of the referencing DiagnosticServiceInstance..
serviceClass	DiagnosticServiceClass	0..1	ref	This represents the corresponding "class", i.e. this meta-class provides properties that are shared among all instances of applicable sub-classes of DiagnosticService Instance. The subclasses that affected by this pattern implement references to the applicable "class"-role that substantiate this abstract reference. **Stereotypes:** atpAbstract

Table A.377: DiagnosticServiceInstance

Class	DiagnosticServiceMappingDiagTarget (abstract)			
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticMapping::ServiceMapping			
Note	This meta-class serves as a base class for diagnostics-related targets of subclasses of DiagnosticSw Mapping			
Base	*ARObject*			
Subclasses	DiagnosticDataElement, DiagnosticParameterElement, DiagnosticParameterIdent			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.378: DiagnosticServiceMappingDiagTarget

Class	DiagnosticServiceSwMapping			
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticMapping::ServiceMapping			
Note	This represents the ability to define a mapping of a diagnostic service to a software-component or a basic-software module. If the former is used then this kind of service mapping is applicable for the usage of ClientServerInterfaces. **Tags:**atp.recommendedPackage=DiagnosticServiceMappings			
Base	*ARElement*, ARObject, CollectableElement, *DiagnosticCommonElement*, DiagnosticMapping, DiagnosticSwMapping, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	DiagnosticServiceSwMapping			
accessedData Prototype	DataPrototype	0..1	iref	Thi instanceRef identifies the DataPrototype that is supposed to be accessed in the context of the operation argument. **InstanceRef implemented by:**DataPrototypeInClient ServerInterfaceInstanceRef
diagnosticData Element	DiagnosticDataElement	0..1	ref	This represents a DiagnosticDataElement required to execute the respective diagnostic service in the context of the diagnostic service mapping,
diagnostic Parameter	DiagnosticParameter Ident	0..1	ref	This represents the applicable payload that corresponds to the referenced DataPrototype in the role mappedData Element.
mappedBsw Service Dependency	BswService DependencyIdent	0..1	ref	This is supposed to represent a reference to a Bsw ServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.
mappedFlatSwc Service Dependency	SwcService Dependency	0..1	ref	This represents the ability to refer to an AtomicSw ComponentType that is available without the definition of how it will be embedded into the component hierarchy.
mappedSwc Service DependencyIn System	SwcService Dependency	0..1	iref	This represents the ability to point into the component hierarchy (under possible consideration of the root SoftwareComposition) **InstanceRef implemented by:**SwcServiceDependency InSystemInstanceRef
parameter ElementAccess	DiagnosticParameter ElementAccess	0..1	aggr	This aggregation represents the single point of access to the reference to one specific DiagnosticParameter Element.
serviceInstance	DiagnosticService Instance	0..1	ref	This represents the service instance that needs to be considered in this diagnostics service mapping.

Table A.379: DiagnosticServiceSwMapping

Class	DiagnosticServiceTable			
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticContribution			
Note	This meta-class represents a model of a diagnostic service table, i.e. the UDS services applicable for a given ECU. **Tags:**atp.recommendedPackage=DiagnosticServiceTables			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	*Mult.*	*Kind*	*Note*
diagnostic Connection	DiagnosticConnection	*	ref	This represents the DiagnosticConnection that is taken for handling the data transmission for the enclosing DiagnosticServiceTable. It is possible to refer to more than one diagnostic Connections in order to support more than one diagnostic tester. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=diagnosticConnection.diagnosticConnection, diagnosticConnection.variationPoint.shortLabel vh.latestBindingTime=postBuild
ecuInstance	EcuInstance	0..1	ref	This represents the applicable EcuInstance for this DiagnosticServiceTable.

▽

△

Class	DiagnosticServiceTable			
protocolKind	NameToken	0..1	attr	This identifies the applicable protocol.
serviceInstance	DiagnosticService Instance	*	ref	This represents the collection of DiagnosticService Instances to be considered in the scope of this Diagnostic ServiceTable,

Table A.380: DiagnosticServiceTable

Class	DiagnosticSession			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm			
Note	This meta-class represents the ability to define a diagnostic session. Tags:atp.recommendedPackage=DiagnosticSessions			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
id	PositiveInteger	0..1	attr	This is the numerical identifier used to identify the DiagnosticSession in the scope of diagnostic workflow
jumpToBoot Loader	DiagnosticJumpToBoot LoaderEnum	0..1	attr	This attribute represents the ability to define whether this diagnostic session allows to jump to Bootloader (OEM Bootloader or System Supplier Bootloader). If this diagnostic session doesn't allow to jump to Bootloader the value JumpToBootLoaderEnum.noBoot shall be chosen.
p2ServerMax	TimeValue	0..1	attr	This is the session value for P2ServerMax in seconds (per Session Control). The AUTOSAR configuration standard is to use SI units, so this parameter is defined as a float value in seconds.
p2StarServer Max	TimeValue	0..1	attr	This is the session value for P2*ServerMax in seconds (per Session Control). The AUTOSAR configuration standard is to use SI units, so this parameter is defined as a float value in seconds.

Table A.381: DiagnosticSession

Class	DiagnosticSessionControl			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::SessionControl			
Note	This represents an instance of the "Session Control" diagnostic service. Tags:atp.recommendedPackage=DiagnosticSessionControls			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
diagnostic Session	DiagnosticSession	0..1	ref	This represents the applicable DiagnosticSessions
sessionControl Class	DiagnosticSession ControlClass	0..1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticSessionControl in the given context.

Table A.382: DiagnosticSessionControl

Class	DiagnosticStartRoutine			
Package	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
Note	This represents the ability to start a diagnostic routine.			
Base	*ARObject*, *DiagnosticRoutineSubfunction*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DiagnosticRoutine.start			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
request	DiagnosticParameter	*	aggr	This represents the request parameters.
response	DiagnosticParameter	*	aggr	This represents the response parameters.

Table A.383: DiagnosticStartRoutine

Class	DiagnosticStopRoutine			
Package	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
Note	This represents the ability to stop a diagnostic routine.			
Base	*ARObject*, *DiagnosticRoutineSubfunction*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	DiagnosticRoutine.stop			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
request	DiagnosticParameter	*	aggr	This represents the request parameters.
response	DiagnosticParameter	*	aggr	This represents the response parameters.

Table A.384: DiagnosticStopRoutine

Class	DiagnosticStorageCondition			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticCondition			
Note	Specification of a storage condition. **Tags:**atp.recommendedPackage=DiagnosticConditions			
Base	*ARElement*, ARObject, CollectableElement, *DiagnosticCommonElement*, *DiagnosticCondition*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.385: DiagnosticStorageCondition

Class	DiagnosticStorageConditionGroup			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticConditionGroup			
Note	Storage condition group which includes one or several storage conditions. **Tags:**atp.recommendedPackage=DiagnosticConditions			
Base	*ARElement*, ARObject, CollectableElement, *DiagnosticCommonElement*, DiagnosticConditionGroup, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

Document ID 635: AUTOSAR_TR_ClassicAutosarModelConstraints

△

Class	DiagnosticStorageConditionGroup			
storage Condition	DiagnosticStorage Condition	1..*	ref	Reference to storageConditions that are part of the StorageConditionGroup. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=storageCondition.diagnosticStorage Condition, storageCondition.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.386: DiagnosticStorageConditionGroup

Class	DiagnosticStorageConditionPortMapping			
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticMapping			
Note	Defines to which SWC service ports with DiagnosticStorageConditionNeeds the DiagnosticStorage Condition is mapped. **Tags:**atp.recommendedPackage=DiagnosticMappings			
Base	*ARElement*, ARObject, CollectableElement, *DiagnosticCommonElement*, DiagnosticMapping, DiagnosticSwMapping, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
diagnostic Storage Condition	DiagnosticStorage Condition	0..1	ref	Reference to the StorageCondition which is mapped to a SWC service port with DiagnosticStorageCondition Needs.
swcFlatService Dependency	SwcService Dependency	0..1	ref	Reference to a SwcServiceDependencyType that links ServiceNeeds to SWC service ports.
swcService DependencyIn System	SwcService Dependency	0..1	iref	Instance reference to a SwcServiceDependency that links ServiceNeeds to SWC service ports. **InstanceRef implemented by:**SwcServiceDependency InSystemInstanceRef

Table A.387: DiagnosticStorageConditionPortMapping

Class	DiagnosticTestIdentifier			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTestResult			
Note	This meta-class represents the ability to create a diagnostic test identifier.			
Base	ARObject			
Aggregated by	DiagnosticTestResult.testIdentifier			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
id	PositiveInteger	1	attr	This represents the numerical id associated with the diagnostic test identifier. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime
uasId	PositiveInteger	1	attr	This represents the unit and scaling Id of the diagnostic test result. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime

Table A.388: DiagnosticTestIdentifier

Class	DiagnosticTestResult			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTestResult			
Note	This meta-class represents the ability to define diagnostic test results. **Tags:**atp.recommendedPackage=DiagnosticTestResults			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
diagnosticEvent	DiagnosticEvent	0..1	ref	This attribute represents the diagnostic event that is related to the diagnostic test result. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=diagnosticEvent.diagnosticEvent, diagnosticEvent.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
monitored Identifier	Diagnostic MeasurementIdentifier	0..1	ref	This attribute represents the related diagnostic monitored identifier.
testIdentifier	DiagnosticTestIdentifier	0..1	aggr	This attribute represents the applicable test identifier.
updateKind	DiagnosticTestResult UpdateEnum	0..1	attr	This attribute controls the update behavior of the enclosing DiagnosticTestResult. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime

Table A.389: DiagnosticTestResult

Class	DiagnosticTestRoutineIdentifier			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x08_RequestControlOfOnBoard Device			
Note	This represents the test id of the DiagnosticTestIdentifier. **Tags:**atp.recommendedPackage=DiagnosticTestRoutineIdentifier			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
id	PositiveInteger	1	attr	This represents the numerical id of the DiagnosticTest Identifier (see SAE J1979-DA).
requestData Size	PositiveInteger	1	attr	This represents the specified data size for the request message. Unit: byte.
responseData Size	PositiveInteger	1	attr	This represents the specified data size for the response message. Unit:byte.

Table A.390: DiagnosticTestRoutineIdentifier

Class	DiagnosticTroubleCodeGroup			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode			
Note	The diagnostic trouble code group defines the DTCs belonging together and thereby forming a group. **Tags:**atp.recommendedPackage=DiagnosticTroubleCodes			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	DiagnosticTroubleCodeGroup			
dtc	DiagnosticTroubleCode	*	ref	This represents the collection of DiagnosticTroubleCodes defined by this DiagnosticTroubleCodeGroup. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=dtc.diagnosticTroubleCode, dtc.variation Point.shortLabel vh.latestBindingTime=postBuild
groupNumber	PositiveInteger	0..1	attr	This represents the base number of the DTC group. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime

Table A.391: DiagnosticTroubleCodeGroup

Class	DiagnosticTroubleCodeJ1939			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode			
Note	This meta-class represents the ability to model specific trouble-code related properties for J1939. **Tags:**atp.recommendedPackage=DiagnosticTroubleCodes			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticTroubleCode*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dtcProps	DiagnosticTroubleCode Props	0..1	ref	Defined properties associated with the J1939 DTC.
fmi	PositiveInteger	0..1	attr	This attribute represents the behavior of the Failure Mode Indicator.
kind	DiagnosticTroubleCode J1939DtcKindEnum	0..1	attr	This attribute further specifies the DTC in terms of its semantics.
node	DiagnosticJ1939Node	0..1	ref	This represents the related DiagnosticJ1939Node.
spn	DiagnosticJ1939Spn	0..1	ref	This represents the releated SPN.

Table A.392: DiagnosticTroubleCodeJ1939

Enumeration	DiagnosticTroubleCodeJ1939DtcKindEnum
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode
Note	This meta-class represents the ability to further specify a J1939 DTC in terms of its semantics.
Aggregated by	DiagnosticTroubleCodeJ1939.kind
Literal	**Description**
serviceOnly	this represents a DTC that is only relevant for service in a garage, reported by e.g. DM53. **Tags:**atp.EnumerationLiteralIndex=0
standard	This represents a non-specific DTC reported by e.g. DM1. **Tags:**atp.EnumerationLiteralIndex=1

Table A.393: DiagnosticTroubleCodeJ1939DtcKindEnum

Class	DiagnosticTroubleCodeObd			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode			
Note	This element is used to define OBD-relevant DTCs. **Tags:**atp.recommendedPackage=DiagnosticTroubleCodes			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticTroubleCode*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
considerPto Status	Boolean	0..1	attr	This attribute describes the affection of the event by the Dem PTO handling. True: the event is affected by the Dem PTO handling. False: the event is not affected by the Dem PTO handling. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime
dtcProps	DiagnosticTroubleCode Props	0..1	ref	Defined properties associated with the DemDTC.
eventReadiness Group	EventObdReadiness Group	0..1	aggr	This aggregation allows for the variant definition of the attribute eventObdReadinessGroup. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=eventReadinessGroup.eventObdReadiness Group, eventReadinessGroup.variationPoint.shortLabel vh.latestBindingTime=postBuild
obdDTCValue	PositiveInteger	0..1	attr	Unique Diagnostic Trouble Code value for OBD. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime
obd DTCValue3Byte	PositiveInteger	0..1	attr	3 Byte OBD DTC value based on the definition from SAE J2012. The existence of this attribute is only required if separated UDS and OBD DTC values are used for SAE J1979-2. If this attribute does not exist, then UDS DTC values are used with J1979-2. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime

Table A.394: DiagnosticTroubleCodeObd

Class	DiagnosticTroubleCodeProps			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode			
Note	This element defines common Dtc properties that can be reused by different non OBD-relevant DTCs. **Tags:**atp.recommendedPackage=DiagnosticTroubleCodePropss			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
aging	DiagnosticAging	0..1	ref	Reference to an aging algorithm in case that an aging/ unlearning of the event is allowed.
diagnostic Memory	DiagnosticMemory Destination	0..1	ref	Reference to the applicable DiagnosticMemory Destination.

▽

△

Class	DiagnosticTroubleCodeProps			
extendedData Record	DiagnosticExtended DataRecord	*	ref	Defines the links to an extended data class sampler. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=extendedDataRecord.diagnosticExtended DataRecord, extendedDataRecord.variationPoint.short Label vh.latestBindingTime=preCompileTime
freezeFrame	DiagnosticFreezeFrame	*	ref	Define the links to a freeze frame class sampler. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=freezeFrame.diagnosticFreezeFrame, freeze Frame.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
immediateNv DataStorage	Boolean	0..1	attr	Switch to enable immediate storage triggering of an according event memory entry persistently to NVRAM. true: immediate non-volatile storage triggering enabled false: immediate non-volatile storage triggering disabled
legislated FreezeFrame ContentUdsObd	DiagnosticDataIdentifier Set	0..1	ref	This reference identifies the layout of legislated freeze frames used for emission related diagnostics over the UDS protocol such as OBDonUDS or WWH-OBD. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=legislatedFreezeFrameContentUds Obd.diagnosticDataIdentifierSet, legislatedFreezeFrame ContentUdsObd.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
maxNumber FreezeFrame Records	PositiveInteger	0..1	attr	This attribute defines the number of according freeze frame records, which can maximal be stored for this event. Therefore all these freeze frame records have the same freeze frame class.
priority	PositiveInteger	0..1	attr	Priority of the event, in view of full event buffer. A lower value means higher priority. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime
significance	DiagnosticSignificance Enum	0..1	attr	Significance of the event, which indicates additional information concerning fault classification and resolution.
snapshot RecordContent	DiagnosticDataIdentifier Set	0..1	ref	This represents the freeze frame layout as a set of DIDs. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=snapshotRecordContent.diagnosticData IdentifierSet, snapshotRecordContent.variationPoint.short Label vh.latestBindingTime=preCompileTime

Table A.395: DiagnosticTroubleCodeProps

Class	DiagnosticTroubleCodeUds
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode
Note	This element is used to describe non OBD-relevant DTCs. **Tags:**atp.recommendedPackage=DiagnosticTroubleCodes
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticTroubleCode*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*
Aggregated by	ARPackage.element

▽

△

Class	DiagnosticTroubleCodeUds			
Attribute	Type	Mult.	Kind	Note
considerPto Status	Boolean	0..1	attr	This attribute describes the affection of the event by the Dem PTO handling. True: the event is affected by the Dem PTO handling. False: the event is not affected by the Dem PTO handling.
dtcProps	DiagnosticTroubleCode Props	0..1	ref	Defined properties associated with the DemDTC.
eventObd Readiness Group	NameToken	0..1	attr	This attribute specifies the Event OBD Readiness group for PID $01 and PID $41 computation. This attribute is only applicable for emission-related ECUs.
functionalUnit	PositiveInteger	0..1	attr	This attribute specifies a 1-byte value which identifies the corresponding basic vehicle / system function which reports the DTC. This parameter is necessary for the report of severity information.
severity	DiagnosticUdsSeverity Enum	0..1	attr	DTC severity according to ISO 14229-1.
udsDtcValue	PositiveInteger	0..1	attr	Unique Diagnostic Trouble Code value for UDS. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime
wwhObdDtc Class	DiagnosticWwhObdDtc ClassEnum	0..1	attr	This attribute is used to identify (if applicable) the corresponding severity class of an WWH-OBD DTC. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime

Table A.396: DiagnosticTroubleCodeUds

Class	DiagnosticValueNeeds			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	Specifies the general needs on the configuration of the Diagnostic Communication Manager (DCM) which are not related to a particular item (e.g. a PID). The main use case is the mapping of service ports to the DCM which are not related to a particular item. In the case of using a sender receiver communicated value, the related value shall be taken via assigned Data in the role "signalBasedDiagnostics". In case of using a client/server communicated value, the related value shall be communicated via the port referenced by assignedPort. The details of this communication (e.g. appropriate naming conventions) are specified in the related software specifications (SWS).			
Base	ARObject, DiagnosticCapabilityElement, Identifiable, MultilanguageReferrable, Referrable, Service Needs			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	Type	Mult.	Kind	Note
dataLength	PositiveInteger	0..1	attr	This attribute is applicable only if the DiagnosticValue Needs is aggregated within a BswModuleDependency. This attribute represents the length of data (in bytes) provided for this particular PID signal.
diagnosticValue Access	DiagnosticValueAccess Enum	0..1	attr	This attribute is applicable only if the DiagnosticValue Needs is aggregated within a BswModuleDependency. This attribute controls whether the data can be read and written or whether it is to be handled read-only.

▽

△

Class	DiagnosticValueNeeds			
fixedLength	Boolean	0..1	attr	This attribute is applicable only if the DiagnosticValue Needs is aggregated within a BswModuleDependency.
				This attribute controls whether the data length of the data is fixed.
processingStyle	DiagnosticProcessing StyleEnum	0..1	attr	This attribute controls whether interaction requires the software-component to react synchronously on a request or whether it processes the request in background but still the DCM has to issue the call again to eventually obtain the result of the request.

Table A.397: DiagnosticValueNeeds

Class	DiagnosticVerifyCertificateBidirectional			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::Authentication			
Note	This meta-class represents the subfunction to do a bidirectional verification of the certificate.			
	Tags:atp.recommendedPackage=DiagnosticAuthentications			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticAuthentication*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.398: DiagnosticVerifyCertificateBidirectional

Class	DiagnosticVerifyCertificateUnidirectional			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::Authentication			
Note	This meta-class represents the subfunction to do a unidirectional verification of the certificate.			
	Tags:atp.recommendedPackage=DiagnosticAuthentications			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticAuthentication*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.399: DiagnosticVerifyCertificateUnidirectional

Class	DiagnosticWriteDataByIdentifier			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::DataByIdentifier			
Note	This represents an instance of the "Write Data by Identifier" diagnostic service.			
	Tags:atp.recommendedPackage=DiagnosticDataByIdentifiers			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticDataByIdentifier*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note

▽

△

Class	DltLogChannel			
segmentation Supported	Boolean	0..1	attr	If enabled, segmentation will be used if a DLT message is larger than Pdu.length referenced via DltLogChannel.tx PduTriggering.
txPduTriggering	PduTriggering	0..1	ref	Reference to DltPdu that is transmitted by the DltLog Channel.

Table A.402: DltLogChannel

Class	DoIpActivationLineNeeds			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	A DoIP entity needs to be informed when an external tester is attached or activated. The DoIpActivation ServiceNeeds specifies the trigger for such an event. Examples would be a Pdu via a regular communication bus, a PWM signal, or an I/O. For details please refer to the ISO 13400.			
Base	ARObject, DoIpServiceNeeds, Identifiable, MultilanguageReferrable, Referrable, ServiceNeeds			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.403: DoIpActivationLineNeeds

Class	DoIpGidNeeds			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	The DoIpGidNeeds indicates that the software-component owning this ServiceNeeds is providing the GID number either after a GID Synchronisation or by other means like e.g. flashed EEPROM parameter. This need can be used independent from DoIpGidSynchronizationNeeds and is necessary if the GID can not be provided out of the DoIP configuration options.			
Base	ARObject, DoIpServiceNeeds, Identifiable, MultilanguageReferrable, Referrable, ServiceNeeds			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.404: DoIpGidNeeds

Class	DoIpGidSynchronizationNeeds			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	The DoIpGidSynchronizationNeeds indicates that the software-component owning this ServiceNeeds is triggered by the DoIP entity to start a synchronization of the GID (Group Identification) on the DoIP service 0x0001, 0x0002, 0x0003 or before announcement via service 0x0004 according to ISO 13400-2:2012 if necessary. Note that this need is only relevant for DoIP synchronization masters.			
Base	ARObject, DoIpServiceNeeds, Identifiable, MultilanguageReferrable, Referrable, ServiceNeeds			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.405: DoIpGidSynchronizationNeeds

Class	DoIpPowerModeStatusNeeds			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	The DoIpPowerModeStatusNeeds indicates that the software-component owning this ServiceNeeds is providing the PowerModeStatus for the DoIP service 0x4003 according to ISO 13400-2:2012.			
Base	*ARObject*, *DoIpServiceNeeds*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *ServiceNeeds*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.406: DoIpPowerModeStatusNeeds

Class	*DoIpServiceNeeds* (abstract)			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	This represents an abstract base class for ServiceNeeds related to DoIP.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *ServiceNeeds*			
Subclasses	DoIpActivationLineNeeds, DoIpGidNeeds, DoIpGidSynchronizationNeeds, DoIpPowerModeStatus Needs, DoIpRoutingActivationAuthenticationNeeds, DoIpRoutingActivationConfirmationNeeds, Further ActionByteNeeds			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.407: DoIpServiceNeeds

Class	DoIpTpConnection			
Package	M2::AUTOSARTemplates::SystemTemplate::DiagnosticConnection			
Note	A connection identifies the sender and the receiver of this particular communication. The DoIp module routes a tpSdu through this connection.			
Base	*ARObject*, *TpConnection*			
Aggregated by	DoIpTpConfig.tpConnection			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
doIpSource Address	DoIpLogicAddress	1	ref	Reference to the address of the sender of the tpSdu.
doIpTarget Address	DoIpLogicAddress	1	ref	Reference to the address of the receiver of the tpSdu.
tpSdu	PduTriggering	1	ref	This reference is used to describe the data exchange between DoIp and the PduR.

Table A.408: DoIpTpConnection

Class	Documentation	
Package	M2::AUTOSARTemplates::GenericStructure::DocumentationOnM1	
Note	This meta-class represents the ability to handle a so called standalone documentation. Standalone means, that such a documentation is not embedded in another ARElement or identifiable object. The standalone documentation is an entity of its own which denotes its context by reference to other objects and instances. **Tags:**atp.recommendedPackage=Documentations	
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*	
Aggregated by	ARPackage.element	

▽

△

Class	Documentation			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
context	DocumentationContext	*	aggr	This is the context of the particular documentation.
documentation Content	PredefinedChapter	0..1	aggr	This is the content of the documentation related to the specified contexts. **Tags:**xml.sequenceOffset=200

Table A.409: Documentation

Class	<<atpMixed>> **DocumentationBlock**			
Package	M2::MSR::Documentation::BlockElements			
Note	This class represents a documentation block. It is made of basic text structure elements which can be displayed in a table cell.			
Base	*ARObject*			
Aggregated by	AUTOSAR.introduction, BlueprintGenerator.introduction, *BlueprintPolicyModifiable*.blueprintDerivation Guide, ClientServerOperationBlueprintMapping.blueprintMappingGuide, *DataMapping*.introduction, Def Item.def, *Describable*.introduction, EcucAddInfoParamValue.value, EcuResourceEstimation.introduction, Entry.entryContents, FrameMapping.introduction, *GeneralAnnotation*.annotationText, *Identifiable*. introduction, IPduMapping.introduction, ISignalMapping.introduction, Item.itemContents, LabeledItem. itemContents, *LifeCycleInfo*.remark, *MappingConstraint*.introduction, MsrQueryP2.msrQueryResultP2, Note.noteText, PortDefinedArgumentBlueprint.blueprintMappingGuide, PrmChar.cond, PrmChar.remark, *ScheduleTableEntry*.introduction, *SignalPathConstraint*.introduction, StructuredReq.conflicts, Structured Req.dependencies, StructuredReq.description, StructuredReq.rationale, StructuredReq.remark, StructuredReq.supportingMaterial, StructuredReq.useCase, SwAxisType.swGenericAxisDesc, Topic Content.blockLevelContent, TraceableText.text, VariationPoint.blueprintCondition			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
defList	DefList	0..1	aggr	This represents a definition list in the documentation block. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=defList, defList.variationPoint.shortLabel vh.latestBindingTime=postBuild xml.sequenceOffset=40
figure	MlFigure	0..1	aggr	This represents a figure in the documentation block. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=figure, figure.variationPoint.shortLabel vh.latestBindingTime=postBuild xml.sequenceOffset=70
formula	MlFormula	0..1	aggr	This is a formula in the definition block. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=formula, formula.variationPoint.shortLabel vh.latestBindingTime=postBuild xml.sequenceOffset=60
labeledList	LabeledList	0..1	aggr	This represents a labeled list. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=labeledList, labeledList.variationPoint.short Label vh.latestBindingTime=postBuild xml.sequenceOffset=50

▽

△

Class	<<atpMixed>> DocumentationBlock			
list	List	0..1	aggr	This represents numbered or unnumbered list. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=list, list.variationPoint.shortLabel vh.latestBindingTime=postBuild xml.sequenceOffset=30
msrQueryP2	MsrQueryP2	0..1	aggr	This represents automatically contributed contents provided by an msrquery in the context of Documentation Block.
note	Note	0..1	aggr	This represents a note in the text flow. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=note, note.variationPoint.shortLabel vh.latestBindingTime=postBuild xml.sequenceOffset=80
p	MultiLanguage Paragraph	0..1	aggr	This is one particular paragraph. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=p, p.variationPoint.shortLabel vh.latestBindingTime=postBuild xml.sequenceOffset=10
structuredReq	StructuredReq	0..1	aggr	This aggregation supports structured requirements embedded in a documentation block. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=structuredReq.shortName, structured Req.variationPoint.shortLabel vh.latestBindingTime=postBuild xml.sequenceOffset=100
trace	TraceableText	0..1	aggr	This represents traceable text in the documentation block. This allows to specify requirements/constraints in any documentation block. The kind of the trace is specified in the category. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=trace.shortName, trace.variationPoint.short Label vh.latestBindingTime=postBuild xml.sequenceOffset=90
verbatim	MultiLanguageVerbatim	0..1	aggr	This represents one particular verbatim text. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=verbatim, verbatim.variationPoint.shortLabel vh.latestBindingTime=postBuild xml.sequenceOffset=20

Table A.410: DocumentationBlock

Class	DocumentationContext
Package	M2::AUTOSARTemplates::GenericStructure::DocumentationOnM1
Note	This class represents the ability to denote a context of a so called standalone documentation. Note that this is an <<atpMixed>>. The contents needs to be considered as ordered.
Base	*ARObject*, *MultilanguageReferrable*, *Referrable*
Aggregated by	Documentation.context

▽

△

Class	DocumentationContext			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
feature	AtpFeature	0..1	iref	This refers to a particular feature (instance in the M0 model) to which is the context of the documentation. **InstanceRef implemented by:**AnyInstanceRef
identifiable	Identifiable	0..1	ref	This is an identifiable object which is part of the context of the documentation.

Table A.411: DocumentationContext

Class	DtcStatusChangeNotificationNeeds			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	This meta-class represents the needs of a software-component interested to get information regarding any DTC status change.			
Base	*ARObject*, *DiagnosticCapabilityElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *Service Needs*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
notificationTime	DiagnosticClearDtc NotificationEnum	0..1	attr	This attribute determines the time when the notification about the DTC operation shall be executed. This attribute is only relevant for the configuration of the ClearDtc Notification.

Table A.412: DtcStatusChangeNotificationNeeds

Class	DynamicPart			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Dynamic part of a multiplexed I-Pdu. Reserved space which is used to transport varying SignalIPdus at the same position, controlled by the corresponding selectorFieldCode.			
Base	*ARObject*, *MultiplexedPart*			
Aggregated by	MultiplexedIPdu.dynamicPart			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dynamicPart Alternative	DynamicPartAlternative	1..*	aggr	Com IPdu alternatives that are transmitted in the Dynamic Part of the MultiplexedIPdu.

Table A.413: DynamicPart

Class	DynamicPartAlternative			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	One of the Com IPdu alternatives that are transmitted in the Dynamic Part of the MultiplexedIPdu. The selectorFieldCode specifies which Com IPdu is contained in the DynamicPart within a certain transmission of a multiplexed PDU.			
Base	*ARObject*			
Aggregated by	DynamicPart.dynamicPartAlternative			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
initialDynamic Part	Boolean	1	attr	Dynamic part that shall be used to initialize this multiplexed IPdu. Constraint: Only one "DynamicPartAlternative" in a "DynamicPart" shall be the initialDynamicPart.

▽

△

Class	DynamicPartAlternative			
iPdu	ISignalIPdu	1	ref	Reference to a Com IPdu which is routed to the IPduM module and is combined to a multiplexedPdu.
selectorField Code	Integer	1	attr	The selector field is part of a multiplexed IPdu. It consists of contiguous bits. The value of the selector field selects the layout of the multiplexed part of the IPdu.

Table A.414: DynamicPartAlternative

Class	E2EProfileCompatibilityProps			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	This meta-class collects settings for configuration of the E2E state machine. **Tags:**atp.recommendedPackage=E2EProfileCompatibilityPropsCollection			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
transitToInvalid Extended	Boolean	0..1	attr	E2E State machine behavior concerning transition from NODATA/INIT to INVALID value=0 (false): no direct transition from NODATA to INVALID, no transition from INIT to INVALID due to counter-related faults (Autosar R19-11 or former behavior) value=1 (true): direct transition from NODATA to INVALID covered, transition from INIT to INVALID due to counter-related faults covered (state machine extended)

Table A.415: E2EProfileCompatibilityProps

Class	ECUMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::ECUResourceMapping			
Note	ECUMapping allows to assign an ECU hardware type (defined in the ECU Resource Template) to an ECUInstance used in a physical topology.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SystemMapping.ecuResourceMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
commController Mapping	Communication ControllerMapping	1..*	aggr	The ECUMapping contains the mapping of all CommunicationControllers of the ECU.
ecu	HwElement	1	ref	Reference to a HwElement of category ECU in the ECU Resource Template.
ecuInstance	EcuInstance	1	ref	Reference to the EcuInstance in the System Template
hwPortMapping	HwPortMapping	1..*	aggr	The ECUMapping contains the mapping of all HW Communication Ports of the ECU.

Table A.416: ECUMapping

Class	EOCEventRef			
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint::ExecutionOrderConstraint			
Note	This is used to define a reference to an RTE or BSW Event.			
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 event is related to.
component	SwComponent Prototype	0..1	iref	This association references the specific instance of the SW-C prototype. **InstanceRef implemented by:**ComponentInComposition InstanceRef
event	AbstractEvent	0..1	ref	The AbstractEvent (event) whose execution order is restricted by the contraint.
successor	EOCExecutableEntity RefAbstract	*	ref	The logical successor of an executable entity or a group of executable entities.

Table A.417: EOCEventRef

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.
component	SwComponent Prototype	0..1	iref	This association references the specific instance of the SW-C prototype. **InstanceRef implemented by:**ComponentInComposition InstanceRef
executable	ExecutableEntity	0..1	ref	The ExecutableEntity whose execution order is restricted by the contraint.
successor	EOCExecutableEntity RefAbstract	*	ref	The logical successor of an executable entity or a group of executable entities.

Table A.418: EOCExecutableEntityRef

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	EOCExecutableEntity RefAbstract	*	ref	The direct successor of an executable entity or a group of executable entities.

Table A.419: 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			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
letInterval	TimingDescriptionEvent Chain	*	ref	This association references the TimingDescriptionEvent Chain that plays the role of a LET interval the executable entities in the group are assigned to.
maxCycles	Integer	0..1	attr	In case of a Repetitive Execution Order Constraint this attribute specifies the number of cycles the Execution Order Constraint is considering.
maxSlots	Integer	0..1	attr	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.
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 A.420: EOCExecutableEntityRefGroup

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	*ARElement*, *ARObject*, *AtomicSwComponentType*, *AtpBlueprint*, *AtpBlueprintable*, *AtpClassifier*, *Atp Type*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *Sw ComponentType*			
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 A.421: EcuAbstractionSwComponentType

Class	EcuInstance			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology			
Note	ECUInstances are used to define the ECUs used in the topology. The type of the ECU is defined by a reference to an ECU specified with the ECU resource description. **Tags:**atp.recommendedPackage=EcuInstances			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			

$\bigtriangledown$

△

Class	**EcuInstance**			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
associatedCom IPduGroup	ISignalIPduGroup	*	ref	With this reference it is possible to identify which ISignal IPduGroups are applicable for which Communication Connector/ ECU. Only top level ISignalIPduGroups shall be referenced by an EcuInstance. If an ISignalIPduGroup contains other ISignalIPduGroups than these contained ISignalIPdu Groups shall not be referenced by the EcuInstance. Contained ISignalIPduGroups are associated to an Ecu Instance via the top level ISignalIPduGroup.
associated Consumed Provided ServiceInstance Group	ConsumedProvided ServiceInstanceGroup	*	ref	With this reference it is possible to identify which ConsumedProvidedServiceInstanceGroups are applicable for which ECUInstance. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=associatedConsumedProvidedService InstanceGroup.consumedProvidedServiceInstanceGroup, associatedConsumedProvidedServiceInstance Group.variationPoint.shortLabel vh.latestBindingTime=postBuild
associatedPdur IPduGroup	PdurIPduGroup	*	ref	With this reference it is possible to identify which PduR IPdu Groups are applicable for which Communication Connector/ ECU.
clientIdRange	ClientIdRange	0..1	aggr	Restriction of the Client Identifier for this Ecu to an allowed range of numerical values. The Client Identifier of the transaction handle is generated by the client RTE for inter-Ecu Client/Server communication.
com Configuration GwTimeBase	TimeValue	0..1	attr	The period between successive calls to Com_Main FunctionRouteSignals of the AUTOSAR COM module in seconds.
com ConfigurationRx TimeBase	TimeValue	0..1	attr	The period between successive calls to Com_Main FunctionRx of the AUTOSAR COM module in seconds.
com ConfigurationTx TimeBase	TimeValue	0..1	attr	The period between successive calls to Com_Main FunctionTx of the AUTOSAR COM module in seconds.
comEnable MDTForCyclic Transmission	Boolean	0..1	attr	Enables for the Com module of this EcuInstance the minimum delay time monitoring for cyclic and repeated transmissions (TransmissionModeTiming has cyclic Timing assigned or eventControlledTiming with numberOf Repetitions > 0).
commController	Communication Controller	1..*	aggr	CommunicationControllers of the ECU. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=commController.shortName, comm Controller.variationPoint.shortLabel vh.latestBindingTime=postBuild
connector	Communication Connector	*	aggr	All channels controlled by a single controller. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=connector.shortName, connector.variation Point.shortLabel vh.latestBindingTime=postBuild
dltConfig	DltConfig	0..1	aggr	Describes the Dlt configuration on this EcuInstance.
doIpConfig	DoIpConfig	0..1	aggr	DoIp configuration on this EcuInstance. **Tags:**atp.Status=draft

▽

△

Class	EcuInstance			
ecuTaskProxy	OsTaskProxy	*	ref	Reference to OsTaskProxies assigned to the Ecu Instance. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=ecuTaskProxy
ethSwitchPort Group Derivation	Boolean	0..1	attr	Defines whether the derivation of SwitchPortGroups based on VLAN and/or CouplingPort.pncMapping shall be performed for this EcuInstance. If not defined the derivation shall not be done.
partition	EcuPartition	*	aggr	Optional definition of Partitions within an Ecu.
pncNmRequest	Boolean	0..1	attr	Defines if this EcuInstance shall request Nm on all its PhysicalChannels which have Nm variant set to FULL each time a PNC is requested.
pncPrepare SleepTimer	TimeValue	0..1	attr	Time in seconds the PNC state machine shall wait in PNC_PREPARE_SLEEP.
pnc Synchronous Wakeup	Boolean	0..1	attr	If this parameter is available and set to true then all available PNCs will be woken up as soon as a channel wakeup occurs. This is ensured by adding all PNCs to all channel wakeup sources during upstream mapping.
pnResetTime	TimeValue	0..1	attr	Specifies the runtime of the reset timer in seconds. This reset time is valid for the reset of PN requests in the EIRA and in the ERA.
sleepMode Supported	Boolean	1	attr	Specifies whether the ECU instance may be put to a "low power mode" • true: sleep mode is supported • false: sleep mode is not supported Note: This flag may only be set to "true" if the feature is supported by both hardware and basic software.
tcpIpIcmpProps	EthTcpIpIcmpProps	0..1	ref	EcuInstance specific ICMP (Internet Control Message Protocol) attributes
tcpIpProps	EthTcpIpProps	0..1	ref	EcuInstance specific TcpIp Stack attributes.
v2xSupported	V2xSupportEnum	0..1	attr	This attribute is used to control the existence of the V2X stack on the given EcuInstance.
wakeUpOver BusSupported	Boolean	1	attr	Driver support for wakeup over Bus.

Table A.422: EcuInstance

Class	EcuPartition			
Package	M2::AUTOSARTemplates::SystemTemplate::SWmapping			
Note	Partitions are used as error containment regions. They permit the grouping of SWCs and resources and allow to describe recovery policies individually for each partition. Partitions can be terminated or restarted during run-time as a result of a detected error.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	EcuInstance.partition			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
execInUser Mode	Boolean	1	attr	A partition can execute either in CPU user mode (execIn UserMode = TRUE) or supervisor mode (execInUser Mode = FALSE). In user mode, the partition has a limited access to memory, to memory mapped hardware and to CPU. In user mode, the partition is mapped to a non-trusted OS-Application.

Table A.423: EcuPartition

Class	EcuResourceEstimation			
Package	M2::AUTOSARTemplates::SystemTemplate::SWmapping			
Note	Resource estimations for RTE and BSW of a single ECU instance.			
Base	*ARObject*			
Aggregated by	SystemMapping.resourceEstimation			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
bswResource Estimation	ResourceConsumption	0..1	aggr	Estimation for the resource consumption of the basic software.
ecuInstance	EcuInstance	1	ref	Reference to the ECU this estimation is done for.
introduction	DocumentationBlock	0..1	aggr	This represents introductory documentation about the ecu resource estimation **Tags:**xml.sequenceOffset=-10
rteResource Estimation	ResourceConsumption	0..1	aggr	Estimation for the resource consumption of the run time environment.
swCompToEcu Mapping	SwcToEcuMapping	*	ref	References to SwcToEcuMappings that have been taken into account for the resource estimations. This way it is possible to define dfferent EcuResourceEstimations with diifferent mappings, e.g. before and after mapping an additional SW component.

Table A.424: EcuResourceEstimation

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	*ARElement*, ARObject, CollectableElement, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *TimingExtension*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
ecu Configuration	EcucValueCollection	0..1	ref	This defines the scope of an EcuTiming. All corresponding timing descriptions and constraints shall be defined within this scope.

Table A.425: EcuTiming

Class	EcucAbstractConfigurationClass (abstract)			
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate			
Note	Specifies the ValueConfigurationClass of a parameter/reference or the MultiplicityConfigurationClass of a parameter/reference or a container for each ConfigurationVariant of the EcucModuleDef.			
Base	*ARObject*			
Subclasses	EcucMultiplicityConfigurationClass, EcucValueConfigurationClass			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
configClass	EcucConfigurationClass Enum	0..1	attr	Specifies the ConfigurationClass for the given ConfigurationVariant.
configVariant	EcucConfiguration VariantEnum	0..1	attr	Specifies the ConfigurationVariant the ConfigurationClass is specified for.

Table A.426: EcucAbstractConfigurationClass

Class	EcucAbstractInternalReferenceDef (abstract)			
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate			
Note	Common abstract class to gather attributes for internal references (where the destination is located in the Ecu Configuration Description).			
Base	ARObject, AtpDefinition, EcucAbstractReferenceDef, EcucCommonAttributes, EcucDefinitionElement, Identifiable, MultilanguageReferrable, Referrable			
Subclasses	EcucChoiceReferenceDef, EcucReferenceDef, EcucUriReferenceDef			
Aggregated by	EcucDestinationUriPolicy.reference, EcucParamConfContainerDef.reference			
Attribute	Type	Mult.	Kind	Note
requires SymbolicName Value	Boolean	0..1	attr	If this attribute is set to true the implementation of the reference is done using a Symbolic Name defined by the referenced container according to TPS_ECUC_02108.

Table A.427: EcucAbstractInternalReferenceDef

Class	EcucAbstractReferenceDef (abstract)			
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate			
Note	Common class to gather the attributes for the definition of references.			
Base	ARObject, AtpDefinition, EcucCommonAttributes, EcucDefinitionElement, Identifiable, Multilanguage Referrable, Referrable			
Subclasses	EcucAbstractExternalReferenceDef, EcucAbstractInternalReferenceDef			
Aggregated by	EcucDestinationUriPolicy.reference, EcucParamConfContainerDef.reference			
Attribute	Type	Mult.	Kind	Note
withAuto	Boolean	0..1	attr	Specifies whether it shall be allowed on the value side to specify this reference value as "AUTO".
				If withAuto is "true" it shall be possible to set the "isAuto Value" attribute of the respective reference to "true". This means that the actual value will not be considered during ECU Configuration but will be (re-)calculated by the code generator and stored in the value attribute afterwards. These implicit updated values might require a re-generation of other modules which reference these values.
				If withAuto is "false" it shall not be possible to set the "is AutoValue" attribute of the respective reference to "true".
				If withAuto is not present the default is "false".

Table A.428: EcucAbstractReferenceDef

Class	EcucAbstractReferenceValue (abstract)			
Package	M2::AUTOSARTemplates::ECUCDescriptionTemplate			
Note	Abstract class to be used as common parent for all reference values in the ECU Configuration Description.			
Base	ARObject, EcucIndexableValue			
Subclasses	EcucInstanceReferenceValue, EcucReferenceValue			
Aggregated by	EcucContainerValue.referenceValue			
Attribute	Type	Mult.	Kind	Note
annotation	Annotation	*	aggr	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.

▽

△

Class	EcucAbstractReferenceValue (abstract)			
definition	EcucAbstractReference Def	0..1	ref	Reference to the definition of this EcucAbstractReference Value subclasses in the ECU Configuration Parameter Definition. **Tags:**xml.sequenceOffset=-10
isAutoValue	Boolean	0..1	attr	If withAuto is set to "true" for this parameter definition the isAutoValue can be set to "true". If isAutoValue is set to "true" the actual value will not be considered during ECU Configuration but will be (re-)calculated by the code generator and stored in the value attribute afterwards. These implicit updated values might require a re-generation of other modules which reference these values. If isAutoValue is not present the default is "false".

Table A.429: EcucAbstractReferenceValue

Class	EcucAddInfoParamValue			
Package	M2::AUTOSARTemplates::ECUCDescriptionTemplate			
Note	This parameter corresponds to EcucAddInfoParamDef.			
Base	ARObject, EcucIndexableValue, EcucParameterValue			
Aggregated by	EcucContainerValue.parameterValue			
Attribute	Type	Mult.	Kind	Note
value	DocumentationBlock	0..1	aggr	Holds the content of the formated text.

Table A.430: EcucAddInfoParamValue

Class	EcucChoiceContainerDef			
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate			
Note	Used to define configuration containers that provide a choice between several EcucParamConfContainer Def. But in the actual ECU Configuration Values only one instance from the choice list will be present.			
Base	ARObject, AtpDefinition, EcucContainerDef, EcucDefinitionElement, Identifiable, Multilanguage Referrable, Referrable			
Aggregated by	EcucDestinationUriPolicy.container, EcucModuleDef.container, EcucParamConfContainerDef.sub Container			
Attribute	Type	Mult.	Kind	Note
choice	EcucParamConf ContainerDef	*	aggr	The choices available in a EcucChoiceContainerDef. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=choice.shortName

Table A.431: EcucChoiceContainerDef

Class	EcucChoiceReferenceDef			
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate			
Note	Specify alternative references where in the ECU Configuration description only one of the specified references will actually be used.			
Base	ARObject, AtpDefinition, EcucAbstractInternalReferenceDef, EcucAbstractReferenceDef, EcucCommon Attributes, EcucDefinitionElement, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	EcucDestinationUriPolicy.reference, EcucParamConfContainerDef.reference			
Attribute	Type	Mult.	Kind	Note

▽

△

Class	EcucChoiceReferenceDef			
destination	EcucContainerDef	*	ref	All the possible parameter containers for the reference are specified. **Stereotypes:** atpUriDef

Table A.432: EcucChoiceReferenceDef

Class	EcucCommonAttributes (abstract)			
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate			
Note	Attributes used by Configuration Parameters as well as References.			
Base	ARObject, AtpDefinition, EcucDefinitionElement, Identifiable, MultilanguageReferrable, Referrable			
Subclasses	EcucAbstractReferenceDef, EcucParameterDef			
Attribute	Type	Mult.	Kind	Note
multiplicity ConfigClass	EcucMultiplicity ConfigurationClass	*	aggr	Specifies in which MultiplicityConfigurationClass this parameter or reference is available in a particular ConfigurationVariant. This aggregation is optional if the surrounding EcucModuleDef has the Category STANDARDIZED_MODULE_DEFINITION. If the category attribute of the EcucModuleDef is set to VENDOR_SPECIFIC_MODULE_DEFINITION, then this aggregation is mandatory. **Tags:**xml.name Plural=MULTIPLICITY-CONFIG-CLASSES
origin	String	0..1	attr	String specifying if this configuration parameter is an AUTOSAR standardized configuration parameter or if the parameter is hardware- or vendor-specific.
postBuildVariant Multiplicity	Boolean	0..1	attr	Indicates if a parameter or a reference may have different number of instances in different post-build variants (previously known as post-build selectable configuration sets). TRUE means yes, FALSE means no.
postBuildVariant Value	Boolean	0..1	attr	Indicates if a parameter or a reference may have different value in different post-build variants (previously known as post-build selectable configuration sets). TRUE means yes, FALSE means no.
requiresIndex	Boolean	0..1	attr	Used to define whether the value element for this definition shall be provided with an index.
valueConfig Class	EcucValueConfiguration Class	*	aggr	Specifies in which ValueConfigurationClass this parameter or reference is available in a particular ConfigurationVariant. This aggregation is optional if the surrounding EcucModuleDef has the Category STANDARDIZED_MODULE_DEFINITION. If the category attribute of the EcucModuleDef is set to VENDOR_SPECIFIC_MODULE_DEFINITION, then this aggregation is mandatory. **Tags:**xml.namePlural=VALUE-CONFIG-CLASSES

Table A.433: EcucCommonAttributes

Class	<<atpMixedString>> EcucConditionFormula
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate
Note	This formula shall yield a boolean expression depending on ecuc queries. Note that the EcucCondition Formula is a mixed string. Therefore, the properties have the upper multiplicity 1.
Base	ARObject, FormulaExpression

▽

△

Class	<<atpMixedString>> **EcucConditionFormula**			
Aggregated by	EcucConditionSpecification.conditionFormula, EcucValidationCondition.validationFormula			
Attribute	***Type***	***Mult.***	***Kind***	***Note***
ecucQuery	EcucQuery	0..1	ref	The EcucQuery serves as a argument for the formula.
ecucQuery String	EcucQuery	0..1	ref	This indicates that the referenced query shall return a string.

Table A.434: EcucConditionFormula

Enumeration	**EcucConfigurationClassEnum**
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate
Note	Possible configuration classes for the AUTOSAR configuration parameters.
Aggregated by	*EcucAbstractConfigurationClass*.configClass
Literal	***Description***
Link	Link Time: parts of configuration are delivered from another object code file **Tags:**atp.EnumerationLiteralIndex=0
PostBuild	PostBuildTime: after compilation a configuration parameter can be changed. **Tags:**atp.EnumerationLiteralIndex=1
PreCompile	PreCompile Time: after compilation a configuration parameter can not be changed any more. **Tags:**atp.EnumerationLiteralIndex=2
Published Information	PublishedInformation is used to specify the fact that certain information is fixed even before the pre-compile stage. **Tags:**atp.EnumerationLiteralIndex=3

Table A.435: EcucConfigurationClassEnum

Enumeration	**EcucConfigurationVariantEnum**
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate
Note	Specifies the possible Configuration Variants used for AUTOSAR BSW Modules.
Aggregated by	*EcucAbstractConfigurationClass*.configVariant, EcucModuleConfigurationValues.implementation ConfigVariant, EcucModuleDef.supportedConfigVariant
Literal	***Description***
Preconfigured Configuration	Preconfigured (i.e. fixed) configuration which cannot be changed. **Tags:**atp.EnumerationLiteralIndex=0
Recommended Configuration	Recommended configuration for a module. **Tags:**atp.EnumerationLiteralIndex=1
VariantLinkTime	Specifies that the BSW Module implementation may use PreCompileTime and LinkTime configuration parameters. **Tags:**atp.EnumerationLiteralIndex=2
VariantPostBuild	Specifies that the BSW Module implementation may use PreCompileTime, LinkTime and PostBuild configuration parameters. **Tags:**atp.EnumerationLiteralIndex=3
VariantPreCompile	Specifies that the BSW Module implementation uses only PreCompileTime configuration parameters. **Tags:**atp.EnumerationLiteralIndex=6

Table A.436: EcucConfigurationVariantEnum

Class	EcucContainerDef (abstract)			
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate			
Note	Base class used to gather common attributes of configuration container definitions.			
Base	ARObject, AtpDefinition, EcucDefinitionElement, Identifiable, MultilanguageReferrable, Referrable			
Subclasses	EcucChoiceContainerDef, EcucParamConfContainerDef			
Aggregated by	EcucDestinationUriPolicy.container, EcucModuleDef.container, EcucParamConfContainerDef.sub Container			
Attribute	Type	Mult.	Kind	Note
destinationUri	EcucDestinationUriDef	*	ref	Several destinationUris can be defined for an Ecuc ContainerDef. With such destinationUris an Ecuc ContainerDef is applicable for several EcucUriReference Defs. **Stereotypes:** atpUriDef
multiplicity ConfigClass	EcucMultiplicity ConfigurationClass	*	aggr	Specifies which MultiplicityConfigurationClass this container is available for which ConfigurationVariant. This aggregation is optional if the surrounding EcucModuleDef has the Category STANDARDIZED_MODULE_ DEFINITION. If the category attribute of the EcucModule Def is set to VENDOR_SPECIFIC_MODULE_ DEFINITION and if the upperMultiplicity is greater than the lowerMultiplicity then this aggregation is mandatory. **Tags:**xml.name Plural=MULTIPLICITY-CONFIG-CLASSES
origin	String	0..1	attr	This attribute specifies whether this configuration container is an AUTOSAR standardized container or whether it is vendor-specific.
postBuildVariant Multiplicity	Boolean	0..1	attr	Indicates if a container may have different number of instances in different post-build variants (previously known as post-build selectable configuration sets). TRUE means yes, FALSE means no.
requiresIndex	Boolean	0..1	attr	Used to define whether the value element for this definition shall be provided with an index.

Table A.437: EcucContainerDef

Class	EcucContainerValue			
Package	M2::AUTOSARTemplates::ECUCDescriptionTemplate			
Note	Represents a Container definition in the ECU Configuration Description.			
Base	ARObject, EcucIndexableValue, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	EcucContainerValue.subContainer, EcucModuleConfigurationValues.container			
Attribute	Type	Mult.	Kind	Note
definition	EcucContainerDef	0..1	ref	Reference to the definition of this Container in the ECU Configuration Parameter Definition. **Tags:**xml.sequenceOffset=-10
parameterValue	EcucParameterValue	*	aggr	Aggregates all ECU Configuration Values within this Container. atpVariation: [RS_ECUC_00079] **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=parameterValue, parameterValue.variation Point.shortLabel vh.latestBindingTime=postBuild

▽

△

Class	EcucContainerValue			
referenceValue	EcucAbstractReference Value	*	aggr	Aggregates all References with this container.
				atpVariation: [RS_ECUC_00079]
				Stereotypes: atpSplitable; atpVariation **Tags:** atp.Splitkey=referenceValue, referenceValue.variation Point.shortLabel vh.latestBindingTime=postBuild
subContainer	EcucContainerValue	*	aggr	Aggregates all sub-containers within this container.
				atpVariation: [RS_ECUC_00078]
				Stereotypes: atpSplitable; atpVariation **Tags:** atp.Splitkey=subContainer.shortName, sub Container.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.438: EcucContainerValue

Class	*EcucDefinitionElement* (abstract)			
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate			
Note	Common class used to express the commonalities of configuration parameters, references and containers. If not stated otherwise the default multiplicity is exactly one mandatory occurrence of the specified element.			
Base	*ARObject*, *AtpDefinition*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	*EcucCommonAttributes*, *EcucContainerDef*, *EcucModuleDef*			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
ecucCond	EcucCondition Specification	0..1	aggr	If it evaluates to true the Ecu Parameter definition shall be processed as specified. Otherwise the parameter definition shall be ignored.
				Tags:xml.sequenceOffset=100
ecucValidation Cond	EcucValidation Condition	*	aggr	Collection of validation conditions which all need to evaluate to true in order to indicate a valid validation condition of the EcucDefinitionElement.
lowerMultiplicity	PositiveInteger	0..1	attr	The lower multiplicity of the specified element.
				0: optional
				1: at least one occurrence
				n: at least n occurrences
				atpVariation: [RS_ECUC_00082]
				Stereotypes: atpVariation **Tags:** vh.latestBindingTime=codeGenerationTime xml.sequenceOffset=110
relatedTrace Item	Traceable	0..1	ref	This contains a sloppy reference to the Autosar compatible identifier of the element (EcucId).
				Stereotypes: atpUriDef **Tags:**xml.sequenceOffset=-10
scope	EcucScopeEnum	0..1	attr	Specifies the scope of this configuration element.
				Tags:xml.sequenceOffset=150

▽

△

Class	EcucDefinitionElement (abstract)			
upperMultiplicity	PositiveInteger	0..1	attr	The upper multiplicity of the specified element.
				0: no occurrence (used for VSMD)
				1: at most one occurrence
				m: at most m occurrences
				If upperMultiplicity is set than upperMultiplicityInfinite shall not be used.
				atpVariation: [RS_ECUC_00082]
				Stereotypes: atpVariation **Tags:** vh.latestBindingTime=codeGenerationTime xml.sequenceOffset=120
upperMultiplicity Infinite	Boolean	0..1	attr	To express an infinite number of occurrences of this element this attribute has to be set to true.
				If upperMultiplicityInfinite is set than upperMultiplicity shall not be used.
				atpVariation: [RS_ECUC_00082]
				Stereotypes: atpVariation **Tags:** vh.latestBindingTime=codeGenerationTime xml.sequenceOffset=130

Table A.439: EcucDefinitionElement

Class	EcucDestinationUriDef			
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate			
Note	Description of an EcucDestinationUriDef that is used as target of EcucUriReferenceDefs.			
Base	ARObject, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcucDestinationUriDefSet.destinationUriDef			
Attribute	Type	Mult.	Kind	Note
destinationUri Policy	EcucDestinationUri Policy	0..1	aggr	Description of the targeted EcucContainerDef.

Table A.440: EcucDestinationUriDef

Class	EcucDestinationUriDefSet			
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate			
Note	This class represents a list of EcucDestinationUriDefs. **Tags:**atp.recommendedPackage=EcucDestinationUriDefSets			
Base	*ARElement*, ARObject, *AtpBlueprint*, AtpBlueprintable, CollectableElement, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
destinationUri Def	EcucDestinationUriDef	*	aggr	This is one particular EcucDestinationUriDef.

Table A.441: EcucDestinationUriDefSet

Enumeration	EcucDestinationUriNestingContractEnum
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate
Note	EcucDestinationUriNestingContractEnum is used to determine what is qualified by the Ecuc DestinationUriPolicy.
Aggregated by	EcucDestinationUriPolicy.destinationUriNestingContract
Literal	Description
leafOfTarget Container	EcucDestinationUriPolicy describes elements (subContainers, Parameters, References) that are directly owned by the target container. **Tags:**atp.EnumerationLiteralIndex=0
targetContainer	EcucDestinationUriPolicy describes the target container of EcucUriReferenceDef. **Tags:**atp.EnumerationLiteralIndex=1
vertexOfTarget Container	EcucDestinationUriPolicy describes elements (subContainers, Parameters, References) of the target container which can be defined in arbitrary nested subContainer structure. **Tags:**atp.EnumerationLiteralIndex=2

Table A.442: EcucDestinationUriNestingContractEnum

Class	EcucDestinationUriPolicy			
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate			
Note	The EcucDestinationUriPolicy describes the EcucContainerDef that will be targeted by EcucUriReference Defs. The type of the description is dependent of the destinationUriNestingContract attribute.			
Base	ARObject			
Aggregated by	EcucDestinationUriDef.destinationUriPolicy			
Attribute	Type	Mult.	Kind	Note
container	EcucContainerDef	*	aggr	Description of the targetContainer in case that the destinationUriNestingPolicy is set to targetContainer. In all other cases the subContainers of the target container are defined here.
destinationUri NestingContract	EcucDestinationUri NestingContractEnum	0..1	attr	This attribute defines how the referenced target Ecuc ContainerDef is described.
parameter	EcucParameterDef	*	aggr	Description of parameters that are contained in the target container.
reference	EcucAbstractReference Def	*	aggr	Description of references that are contained in the target container.

Table A.443: EcucDestinationUriPolicy

Class	EcucEnumerationLiteralDef			
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate			
Note	Configuration parameter type for enumeration literals definition.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	EcucEnumerationParamDef.literal			
Attribute	Type	Mult.	Kind	Note
ecucCond	EcucCondition Specification	0..1	aggr	If it evaluates to true the literal definition shall be processed as specified. Otherwise the literal definition shall be ignored.
origin	String	0..1	attr	String specifying if this literal is an AUTOSAR standardized literal or if the literal is vendor-specific.

Table A.444: EcucEnumerationLiteralDef

Class	EcucForeignReferenceDef			
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate			
Note	Specify a reference to an XML description of an entity described in another AUTOSAR template.			
Base	*ARObject*, *AtpDefinition*, *EcucAbstractExternalReferenceDef*, *EcucAbstractReferenceDef*, *Ecuc CommonAttributes*, *EcucDefinitionElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcucDestinationUriPolicy.reference, EcucParamConfContainerDef.reference			
Attribute	Type	Mult.	Kind	Note
destinationType	String	0..1	attr	The type in the AUTOSAR Metamodel to which instance this reference is allowed to point to.

Table A.445: EcucForeignReferenceDef

Class	<<atpVariation>> **EcucFunctionNameDef**			
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate			
Note	Configuration parameter type for Function Names like those used to specify callback functions.			
Base	*ARObject*, *AtpDefinition*, *EcucAbstractStringParamDef*, *EcucCommonAttributes*, *EcucDefinitionElement*, *EcucParameterDef*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcucDestinationUriPolicy.parameter, EcucParamConfContainerDef.parameter			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.446: EcucFunctionNameDef

Class	EcucInstanceReferenceDef			
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate			
Note	Specify a reference to an XML description of an entity described in another AUTOSAR template using the INSTANCE REFERENCE semantics.			
Base	*ARObject*, *AtpDefinition*, *EcucAbstractExternalReferenceDef*, *EcucAbstractReferenceDef*, *Ecuc CommonAttributes*, *EcucDefinitionElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcucDestinationUriPolicy.reference, EcucParamConfContainerDef.reference			
Attribute	Type	Mult.	Kind	Note
destination Context	String	0..1	attr	The context in the AUTOSAR Metamodel to which' this reference is allowed to point to.
destinationType	String	0..1	attr	The type in the AUTOSAR Metamodel to which' instance this reference is allowed to point to.

Table A.447: EcucInstanceReferenceDef

Class	EcucInstanceReferenceValue			
Package	M2::AUTOSARTemplates::ECUCDescriptionTemplate			
Note	InstanceReference representation in the ECU Configuration.			
Base	*ARObject*, *EcucAbstractReferenceValue*, *EcucIndexableValue*			
Aggregated by	EcucContainerValue.referenceValue			
Attribute	Type	Mult.	Kind	Note
value	AtpFeature	0..1	iref	InstanceReference representation in the ECU Configuration. **InstanceRef implemented by:**AnyInstanceRef

Table A.448: EcucInstanceReferenceValue

Class	EcucModuleConfigurationValues
Package	M2::AUTOSARTemplates::ECUCDescriptionTemplate
Note	Head of the configuration of one Module. A Module can be a BSW module as well as the RTE and ECU Infrastructure.
	As part of the BSW module description, the EcucModuleConfigurationValues element has two different roles:
	The recommendedConfiguration contains parameter values recommended by the BSW module vendor.
	The preconfiguredConfiguration contains values for those parameters which are fixed by the implementation and cannot be changed.
	These two EcucModuleConfigurationValues are used when the base EcucModuleConfigurationValues (as part of the base ECU configuration) is created to fill parameters with initial values.
	Tags:atp.recommendedPackage=EcucModuleConfigurationValuess
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*
Aggregated by	ARPackage.element

Attribute	Type	Mult.	Kind	Note
container	EcucContainerValue	*	aggr	Aggregates all containers that belong to this module configuration. atpVariation: [RS_ECUC_00078] **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=container.shortName, container.variation Point.shortLabel vh.latestBindingTime=postBuild xml.sequenceOffset=10
definition	EcucModuleDef	0..1	ref	Reference to the definition of this EcucModule ConfigurationValues element. Typically, this is a vendor specific module configuration. **Tags:**xml.sequenceOffset=-10
ecucDefEdition	RevisionLabelString	0..1	attr	This is the version info of the ModuleDef ECUC Parameter definition to which this values conform to / are based on. For the Definition of ModuleDef ECUC Parameters the AdminData shall be used to express the semantic changes. The compatibility rules between the definition and value revision labels is up to the module's vendor.
implementation ConfigVariant	EcucConfiguration VariantEnum	0..1	attr	Specifies the kind of deliverable this EcucModule ConfigurationValues element provides. If this element is not used in a particular role (e.g. preconfigured Configuration or recommendedConfiguration) then the value shall be one of VariantPreCompile, VariantLink Time, VariantPostBuild.
module Description	BswImplementation	0..1	ref	Referencing the BSW module description, which this EcucModuleConfigurationValues element is configuring. This is optional because the EcucModuleConfiguration Values element is also used to configure the ECU infrastructure (memory map) or Application SW-Cs. However in case the EcucModuleConfigurationValues are used to configure the module, the reference is mandatory in order to fetch module specific "common" published information.
postBuildVariant Used	Boolean	0..1	attr	Indicates whether a module implementation has or plans to have (i.e., introduced at link or post-build time) new post-build variation points. TRUE means yes, FALSE means no. If the attribute is not defined, FALSE semantics shall be assumed.

Table A.449: EcucModuleConfigurationValues

Class	EcucModuleDef			
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate			
Note	Used as the top-level element for configuration definition for Software Modules, including BSW and RTE as well as ECU Infrastructure. **Tags:**atp.recommendedPackage=EcucModuleDefs			
Base	*ARElement*, ARObject, *AtpBlueprint*, AtpBlueprintable, AtpDefinition, CollectableElement, *Ecuc DefinitionElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
apiServicePrefix	CIdentifier	0..1	attr	For modules where several instances of the VSMD can be defined the apiServicePrefix defines the API namespace of the derived instances, e.g. Cdd, Xfrm (ComXf, SomeIpXf, E2EXf).
container	EcucContainerDef	*	aggr	Aggregates the top-level container definitions of this specific module definition. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=container.shortName xml.sequenceOffset=11
postBuildVariant Support	Boolean	0..1	attr	Indicates if a module supports different post-build variants (previously known as post-build selectable configuration sets). TRUE means yes, FALSE means no.
refinedModule Def	EcucModuleDef	0..1	ref	Optional reference from the Vendor Specific Module Definition to the Standardized Module Definition it refines. In case this EcucModuleDef has the category STANDARDIZED_MODULE_DEFINITION this reference shall not be provided. In case this EcucModuleDef has the category VENDOR_SPECIFIC_MODULE_ DEFINITION this reference is mandatory. **Stereotypes:** atpUriDef
supported ConfigVariant	EcucConfiguration VariantEnum	*	attr	Specifies which ConfigurationVariants are supported by this software module. This attribute is optional if the Ecuc ModuleDef has the category STANDARDIZED_ MODULE_DEFINITION. If the category attribute of the EcucModuleDef is set to VENDOR_SPECIFIC_ MODULE_DEFINITION then this attribute is mandatory.

Table A.450: EcucModuleDef

Class	EcucMultiplicityConfigurationClass			
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate			
Note	Specifies the MultiplicityConfigurationClass of a parameter/reference or a container for each ConfigurationVariant of the EcucModuleDef.			
Base	ARObject, *EcucAbstractConfigurationClass*			
Aggregated by	*EcucCommonAttributes*.multiplicityConfigClass, *EcucContainerDef*.multiplicityConfigClass			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.451: EcucMultiplicityConfigurationClass

Class	EcucNumericalParamValue			
Package	M2::AUTOSARTemplates::ECUCDescriptionTemplate			
Note	Holding the value which is subject to variant handling.			
Base	ARObject, EcucIndexableValue, EcucParameterValue			
Aggregated by	EcucContainerValue.parameterValue			
Attribute	Type	Mult.	Kind	Note
value	Numerical	0..1	attr	Value which is subject to variant handling. atpVariation: [RS_ECUC_00080] **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime

Table A.452: EcucNumericalParamValue

Class	EcucParamConfContainerDef			
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate			
Note	Used to define configuration containers that can hierarchically contain other containers and/or parameter definitions.			
Base	ARObject, AtpDefinition, EcucContainerDef, EcucDefinitionElement, Identifiable, Multilanguage Referrable, Referrable			
Aggregated by	EcucChoiceContainerDef.choice, EcucDestinationUriPolicy.container, EcucModuleDef.container, Ecuc ParamConfContainerDef.subContainer			
Attribute	Type	Mult.	Kind	Note
parameter	EcucParameterDef	*	aggr	The parameters defined within the EcucParamConf ContainerDef. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=parameter.shortName
reference	EcucAbstractReference Def	*	aggr	The references defined within the EcucParamConf ContainerDef. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=reference.shortName
subContainer	EcucContainerDef	*	aggr	The containers defined within the EcucParamConf ContainerDef. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=subContainer.shortName

Table A.453: EcucParamConfContainerDef

Class	EcucParameterDef (abstract)			
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate			
Note	Abstract class used to define the similarities of all ECU Configuration Parameter types defined as subclasses.			
Base	ARObject, AtpDefinition, EcucCommonAttributes, EcucDefinitionElement, Identifiable, Multilanguage Referrable, Referrable			
Subclasses	EcucAbstractStringParamDef, EcucAddInfoParamDef, EcucBooleanParamDef, EcucEnumerationParam Def, EcucFloatParamDef, EcucIntegerParamDef			
Aggregated by	EcucDestinationUriPolicy.parameter, EcucParamConfContainerDef.parameter			
Attribute	Type	Mult.	Kind	Note
derivation	EcucDerivation Specification	0..1	aggr	A derivation of a Configuration Parameter value can be specified by an informal Calculation Formula or by a formal language that can be used to specify the computational rules.

▽

△

Class	EcucParameterDef (abstract)			
symbolicName Value	Boolean	0..1	attr	Specifies that this parameter's value is used, together with the aggregating container, to derive a symbolic name definition. See chapter "Representation of Symbolic Names" in Ecuc specification for more details.
withAuto	Boolean	0..1	attr	Specifies whether it shall be allowed on the value side to specify this parameter value as "AUTO". If withAuto is "true" it shall be possible to set the "isAuto Value" attribute of the respective parameter to "true". This means that the actual value will not be considered during ECU Configuration but will be (re-)calculated by the code generator and stored in the value attribute afterwards. These implicit updated values might require a re-generation of other modules which reference these values. If withAuto is "false" it shall not be possible to set the "is AutoValue" attribute of the respective parameter to "true". If withAuto is not present the default is "false".

Table A.454: EcucParameterDef

Class	EcucParameterValue (abstract)			
Package	M2::AUTOSARTemplates::ECUCDescriptionTemplate			
Note	Common class to all types of configuration values.			
Base	ARObject, EcucIndexableValue			
Subclasses	EcucAddInfoParamValue, EcucNumericalParamValue, EcucTextualParamValue			
Aggregated by	EcucContainerValue.parameterValue			
Attribute	Type	Mult.	Kind	Note
annotation	Annotation	*	aggr	Possibility to provide additional notes while defining the ECU Configuration Parameter Values. These are not intended as documentation but are mere design notes. **Tags:**xml.sequenceOffset=10
definition	EcucParameterDef	0..1	ref	Reference to the definition of this EcucParameterValue subclasses in the ECU Configuration Parameter Definition. **Tags:**xml.sequenceOffset=-10
isAutoValue	Boolean	0..1	attr	If withAuto is set to "true" for this parameter definition the isAutoValue can be set to "true". If isAutoValue is set to "true" the actual value will not be considered during ECU Configuration but will be (re-)calculated by the code generator and stored in the value attribute afterwards. These implicit updated values might require a re-generation of other modules which reference these values. If isAutoValue is not present the default is "false". **Tags:**xml.sequenceOffset=20

Table A.455: EcucParameterValue

Class	EcucQuery			
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate			
Note	Defines a query to the ECUC Description.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcucConditionSpecification.ecucQuery, EcucDerivationSpecification.ecucQuery, EcucValidation Condition.ecucQuery			
Attribute	**Type**	*Mult.*	**Kind**	**Note**
ecucQuery Expression	EcucQueryExpression	0..1	aggr	This is the EcucQuery used in the calculation formula or the condition formula.

Table A.456: EcucQuery

Class	<<atpMixedString>> **EcucQueryExpression**			
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate			
Note	Defines a query expression to the ECUC Description and output the result as an numerical value. Due to the "mixedString" nature of the formula there can be several EcuQueryExpressions used.			
Base	*ARObject*			
Aggregated by	EcucQuery.ecucQueryExpression			
Attribute	**Type**	*Mult.*	**Kind**	**Note**
configElement DefGlobal	EcucDefinitionElement	0..1	ref	The EcucQueryExpression points to an EcucDefinition Element that is used to find an element in the Ecuc Description. In order to find the right element in the Ecuc Description a search is necessary. If the complete Ecuc Description needs to be searched this global reference shall be used. Due to the "mixedString" nature of the EcucQueryExpression several references to Ecuc DefintionElements can be used in one EcucQuery Expression. **Stereotypes:** atpUriDef
configElement DefLocal	EcucDefinitionElement	0..1	ref	The EcucQueryExpression points to an EcucDefinition Element that is used to find an element in the Ecuc Description. In order to find the right element in the Ecuc Description a search is necessary. If the search is executed inside of the same module that contains the EcucQuery this local reference shall be used. Due to the "mixedString" nature of the EcucQueryExpression several references to EcucDefintionElements can be used in one EcucQueryExpression. **Stereotypes:** atpUriDef

Table A.457: EcucQueryExpression

Class	EcucReferenceDef			
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate			
Note	Specify references within the ECU Configuration Description between parameter containers.			
Base	*ARObject*, *AtpDefinition*, *EcucAbstractInternalReferenceDef*, *EcucAbstractReferenceDef*, *EcucCommon Attributes*, *EcucDefinitionElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcucDestinationUriPolicy.reference, EcucParamConfContainerDef.reference			
Attribute	**Type**	*Mult.*	**Kind**	**Note**
destination	EcucContainerDef	0..1	ref	Exactly one reference to a parameter container is allowed as destination. **Stereotypes:** atpUriDef

Table A.458: EcucReferenceDef

Class	EcucReferenceValue			
Package	M2::AUTOSARTemplates::ECUCDescriptionTemplate			
Note	Used to represent a configuration value that has a parameter definition of type EcucAbstractReference Def (used for all of its specializations excluding EcucInstanceReferenceDef).			
Base	*ARObject*, *EcucAbstractReferenceValue*, *EcucIndexableValue*			
Aggregated by	EcucContainerValue.referenceValue			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
value	Referrable	0..1	ref	Specifies the destination of the reference.

Table A.459: EcucReferenceValue

Class	EcucTextualParamValue			
Package	M2::AUTOSARTemplates::ECUCDescriptionTemplate			
Note	Holding a value which is not subject to variation.			
Base	*ARObject*, *EcucIndexableValue*, *EcucParameterValue*			
Aggregated by	EcucContainerValue.parameterValue			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
value	VerbatimString	0..1	attr	Value of the parameter, not subject to variant handling.

Table A.460: EcucTextualParamValue

Class	EcucUriReferenceDef			
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate			
Note	Definition of reference with a destination that is specified via a destinationUri. With such a reference it is possible to define a reference to a EcucContainerDef in a different module independent from the concrete definition of the target container.			
Base	*ARObject*, *AtpDefinition*, *EcucAbstractInternalReferenceDef*, *EcucAbstractReferenceDef*, *EcucCommon Attributes*, *EcucDefinitionElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcucDestinationUriPolicy.reference, EcucParamConfContainerDef.reference			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
destinationUri	EcucDestinationUriDef	0..1	ref	Any EcucContainerDef with a destinationUri that is identical to the destinationUri that is referenced here defines a valid target. **Stereotypes:** atpUriDef

Table A.461: EcucUriReferenceDef

Class	EcucValidationCondition			
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate			
Note	Validation condition to perform a formula calculation based on EcucQueries.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*EcucDefinitionElement*.ecucValidationCond			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
ecucQuery	EcucQuery	*	aggr	Query to the ECU Configuration Description.
validation Formula	EcucConditionFormula	0..1	aggr	Definition of the formula used to define validation condition.

Table A.462: EcucValidationCondition

Class	EcucValueCollection			
Package	M2::AUTOSARTemplates::ECUCDescriptionTemplate			
Note	This represents the anchor point of the ECU configuration description. **Tags:**atp.recommendedPackage=EcucValueCollections			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
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:** atpSplitable; 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 A.463: EcucValueCollection

Class	EcucValueConfigurationClass			
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate			
Note	Specifies the ValueConfigurationClass of a parameter/reference for each ConfigurationVariant of the EcucModuleDef.			
Base	*ARObject*, *EcucAbstractConfigurationClass*			
Aggregated by	*EcucCommonAttributes*.valueConfigClass			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.464: EcucValueConfigurationClass

Class	EndToEndDescription			
Package	M2::AUTOSARTemplates::SWComponentTemplate::EndToEndProtection			
Note	This meta-class contains information about end-to-end protection. The set of applicable attributes depends on the actual value of the category attribute of EndToEndProtection.			
Base	*ARObject*			
Aggregated by	EndToEndProtection.endToEndProfile			
Attribute	Type	Mult.	Kind	Note
category	NameToken	0..1	attr	The category represents the identification of the concrete E2E profile. The applicable values are specified in a semantic constraint and determine the applicable attributes of EndToEndDescription. **Tags:**xml.sequenceOffset=-100

$\bigtriangledown$

△

Class	**EndToEndDescription**			
counterOffset	PositiveInteger	0..1	attr	Bit offset of Counter from the beginning of the Array representation of the Signal Group/VariableDataPrototype (MSB order, bit numbering: bit 0 is the least important). The offset shall be a multiplicity of 4 and it should be 8 whenever possible. For example, offset 8 means that the counter will take the low nibble of the byte 1, i.e. bits 8 .. 11. If counterOffset is not present the value is defined by the selected profile. **Tags:**xml.sequenceOffset=-50
crcOffset	PositiveInteger	0..1	attr	Bit offset of CRC from the beginning of the Array representation of the Signal Group/VariableDataPrototype (MSB order, bit numbering: bit 0 is the least important). The offset shall be a multiplicity of 8 and it should be 0 whenever possible. For example, offset 8 means that the CRC will take the byte 1, i.e. bits 8..15. If crcOffset is not present the value is defined by the selected profile. **Tags:**xml.sequenceOffset=-60
dataId (ordered)	PositiveInteger	*	attr	This represents a unique numerical identifier. Note: ID is used for protection against masquerading. The details concerning the maximum number of values (this information is specific for each E2E profile) applicable for this attribute are controlled by a semantic constraint that depends on the category of the EndToEnd Protection. **Tags:**xml.sequenceOffset=-90
dataIdMode	PositiveInteger	0..1	attr	There are three inclusion modes how the implicit two-byte Data ID is included in the one-byte CRC: • dataIDMode = 0: Two bytes are included in the CRC (double ID configuration) This is used in variant 1A. • dataIDMode = 1: One of the two bytes byte is included, alternating high and low byte, depending on parity of the counter (alternating ID configuration). For even counter low byte is included; For odd counters the high byte is included. This is used in variant 1B. • dataIDMode = 2: Only low byte is included, high byte is never used. This is applicable if the IDs in a particular system are 8 bits. • dataIdMode = 3: The low byte is included in the implicit CRC calculation, the low nibble of the high byte is transmitted along with the data (i.e. it is explicitly included), the high nibble of the high byte is not used. This is applicable for the IDs up to 12 bits. **Tags:**xml.sequenceOffset=-85
dataIdNibble Offset	PositiveInteger	0..1	attr	Bit offset of the low nibble of the high byte of Data ID. The applicability of this attribute is controlled by [constr_1261]. **Tags:**xml.sequenceOffset=-25
dataLength	PositiveInteger	0..1	attr	This attribute represents the length of the Array representation of the Signal Group/VariableDataPrototype including CRC and Counter in bits. **Tags:**xml.sequenceOffset=-80

▽

△

Class	EndToEndDescription			
maxDelta CounterInit	PositiveInteger	0..1	attr	Initial maximum allowed gap between two counter values of two consecutively received valid Data, i.e. how many subsequent lost data is accepted. For example, if the receiver gets Data with counter 1 and MaxDeltaCounter Init is 1, then at the next reception the receiver can accept Counters with values 2 and 3, but not 4.
				Note that if the receiver does not receive new Data at a consecutive read, then the receiver increments the tolerance by 1.
				Tags:xml.sequenceOffset=-70
maxNoNewOr RepeatedData	PositiveInteger	0..1	attr	The maximum amount of missing or repeated Data which the receiver does not expect to exceed under normal communication conditions.
				Tags:xml.sequenceOffset=-40
syncCounterInit	PositiveInteger	0..1	attr	Number of Data required for validating the consistency of the counter that shall be received with a valid counter (i.e. counter within the allowed lock-in range) after the detection of an unexpected behavior of a received counter.
				Tags:xml.sequenceOffset=-30

Table A.465: EndToEndDescription

Enumeration	EndToEndProfileBehaviorEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer
Note	Behavior of the check functionality
Aggregated by	EndToEndTransformationDescription.profileBehavior
Literal	Description
PRE_R4_2	Check has the legacy behavior, before AUTOSAR Release 4.2.
	Tags:atp.EnumerationLiteralIndex=0
R4_2	Check behaves like new P4/P5/P6 profiles introduced in AUTOSAR Release 4.2.
	Tags:atp.EnumerationLiteralIndex=1

Table A.466: EndToEndProfileBehaviorEnum

Class	EndToEndProtection			
Package	M2::AUTOSARTemplates::SWComponentTemplate::EndToEndProtection			
Note	This meta-class represents the ability to describe a particular end to end protection.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EndToEndProtectionSet.endToEndProtection			
Attribute	Type	Mult.	Kind	Note
endToEnd Profile	EndToEndDescription	0..1	aggr	This represents the particular EndToEndDescription.
				Stereotypes: atpSplitable
				Tags:atp.Splitkey=endToEndProfile

▽

△

Class	EndToEndProtection				
endToEnd Protection ISignalIPdu	EndToEndProtectionI SignalIPdu	*	aggr		Defines to which ISignalIPdu - ISignalGroup pair this End ToEndProtection shall apply. In case several ISignalGroups are used to transport the data (e.g. fan-out in the RTE) there may exist several End ToEndProtectionISignalIPdu definitions. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=endToEndProtectionISignalIPdu, endToEnd ProtectionISignalIPdu.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
endToEnd Protection Variable Prototype	EndToEndProtection VariablePrototype	*	aggr		Defines to which VariableDataPrototypes in the roles of one sender and one or more receivers this EndTo Endprotection applies. It shall be possible to aggregate several EndToEnd ProtectionVariablePrototype in case additional hierarchical decompositions are introduced subsequently. In this case one particular PortPrototype is split into multiple PortPrototypes and connectors, all representing the same data entity. Caveat: The E2E wrapper approach involves technologies that are not subjected to the AUTOSAR standard and is superseded by the superior E2E transformer approach (which is fully standardized by AUTOSAR). Hence, new projects (without legacy constraints due to carry-over parts) shall use the fully standardized E2E transformer approach. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=endToEndProtectionVariablePrototype.short Label, endToEndProtectionVariablePrototype.variation Point.shortLabel vh.latestBindingTime=preCompileTime

Table A.467: EndToEndProtection

Class	EndToEndProtectionVariablePrototype				
Package	M2::AUTOSARTemplates::SWComponentTemplate::EndToEndProtection				
Note	It is possible to protect the data exchanged between software components. For this purpose, for each communication to be protected, the user defines a separate EndToEndProtection (specifying a set of protection settings) and refers to a variableDataPrototype in the role of sender and to one or many variableDataPrototypes in the role of receiver. For details, see EndToEnd Library. Caveat: The E2E wrapper approach involves technologies that are not subjected to the AUTOSAR standard and is superseded by the superior E2E transformer approach (which is fully standardized by AUTOSAR). Hence, new projects (without legacy constraints due to carry-over parts) shall use the fully standardized E2E transformer approach.				
Base	ARObject				
Aggregated by	EndToEndProtection.endToEndProtectionVariablePrototype				
Attribute	**Type**	**Mult.**	**Kind**	**Note**	
receiver	VariableDataPrototype	*	iref	This represents the receiver. Note that 1:n communication is supported for this use case. **InstanceRef implemented by:**VariableDataPrototypeIn SystemInstanceRef	

▽

△

Class	EndToEndProtectionVariablePrototype			
sender	VariableDataPrototype	0..1	iref	This represents the sender.
				Can be optional if an ecu extract is provided and the sender is part of the extract.
				InstanceRef implemented by: VariableDataPrototypeInSystemInstanceRef
shortLabel	Identifier	0..1	attr	This serves as part of the split key in case of more than one EndToEndProtectionVariablePrototype is aggregated in the bound model.
				Stereotypes: atpIdentityContributor

Table A.468: EndToEndProtectionVariablePrototype

Class	EndToEndTransformationComSpecProps			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	The class EndToEndTransformationIComSpecProps specifies port specific configuration properties for EndToEnd transformer attributes.			
Base	*ARObject*, *Describable*, *TransformationComSpecProps*			
Aggregated by	ClientComSpec.transformationComSpecProps, *ReceiverComSpec*.transformationComSpecProps, ServerComSpec.transformationComSpecProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
clearFromValidToInvalid	Boolean	0..1	attr	Clear monitoring window on transition from state Valid to state Invalid.
disableEndToEndCheck	Boolean	0..1	attr	Disables/Enables the E2E check. The E2Eheader is removed from the payload independent from the setting of this attribute.
disableEndToEndStateMachine	Boolean	0..1	attr	Disables the E2EStateMachine (only E2E check functionality is performed)
e2eProfileCompatibilityProps	E2EProfileCompatibilityProps	0..1	ref	Reference to additional settings for the E2E state machine.
maxDeltaCounter	PositiveInteger	0..1	attr	Maximum allowed difference between two counter values of two consecutively received valid messages. For example, if the receiver gets data with counter 1 and Max DeltaCounter is 3, then at the next reception the receiver can accept Counters with values 2, 3 or 4.
maxErrorStateInit	PositiveInteger	0..1	attr	Maximal number of checks in which ProfileStatus equal to E2E_P_ERROR was determined, within the last Window Size checks, for the state E2E_SM_INIT. The minimum value is 0.
maxErrorStateInvalid	PositiveInteger	0..1	attr	Maximal number of checks in which ProfileStatus equal to E2E_P_ERROR was determined, within the last Window Size checks, for the state E2E_SM_INVALID. The minimum value is 0.
maxErrorStateValid	PositiveInteger	0..1	attr	Maximal number of checks in which ProfileStatus equal to E2E_P_ERROR was determined, within the last Window Size checks, for the state E2E_SM_VALID. The minimum value is 0.
maxNoNewOrRepeatedData	PositiveInteger	0..1	attr	EndToEndTransformationDescription holds these attributes which are profile specific and have the same value for all E2E transformers.

▽

△

Class	**EndToEndTransformationComSpecProps**			
minOkStateInit	PositiveInteger	0..1	attr	Minimal number of checks in which ProfileStatus equal to E2E_P_OK was determined, within the last WindowSize checks, for the state E2E_SM_INIT. The minimum value is 1.
minOkState Invalid	PositiveInteger	0..1	attr	Minimal number of checks in which ProfileStatus equal to E2E_P_OK was determined, within the last WindowSize checks, for the state E2E_SM_INVALID. The minimum value is 1.
minOkState Valid	PositiveInteger	0..1	attr	Minimal number of checks in which ProfileStatus equal to E2E_P_OK was determined, within the last WindowSize checks, for the state E2E_SM_VALID. The minimum value is 1.
syncCounterInit	PositiveInteger	0..1	attr	EndToEndTransformationDescription holds these attributes which are profile specific and have the same value for all E2E transformers.
windowSizeInit	PositiveInteger	0..1	attr	Size of the monitoring window of state Init for the E2E state machine.
windowSize Invalid	PositiveInteger	0..1	attr	Size of the monitoring window of state Invalid for the E2E state machine.
windowSize Valid	PositiveInteger	0..1	attr	Size of the monitoring window of state Valid for the E2E state machine.

Table A.469: EndToEndTransformationComSpecProps

Class	**EndToEndTransformationDescription**			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	EndToEndTransformationDescription holds these attributes which are profile specific and have the same value for all E2E transformers.			
Base	*ARObject*, *Describable*, *TransformationDescription*			
Aggregated by	TransformationTechnology.transformationDescription			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
clearFromValid ToInvalid	Boolean	0..1	attr	Clear monitoring window on transition from state Valid to state Invalid.
counterOffset	PositiveInteger	0..1	attr	Offset of the counter in the Data[] array in bits.
crcOffset	PositiveInteger	0..1	attr	Offset of the CRC in the Data[] array in bits.
dataIdMode	DataIdModeEnum	0..1	attr	This attribute describes the inclusion mode that is used to include the implicit two-byte Data ID in the one-byte CRC.
dataIdNibble Offset	PositiveInteger	0..1	attr	Offset of the Data ID nibble in the Data[] array in bits.
e2eProfile Compatibility Props	E2EProfileCompatibility Props	0..1	ref	Reference to additional settings for the E2E state machine.
maxDelta Counter	PositiveInteger	0..1	attr	Maximum allowed difference between two counter values of two consecutively received valid messages. For example, if the receiver gets data with counter 1 and Max DeltaCounter is 3, then at the next reception the receiver can accept Counters with values 2, 3 or 4.
maxErrorState Init	PositiveInteger	0..1	attr	Maximal number of checks in which ProfileStatus equal to E2E_P_ERROR was determined, within the last Window Size checks, for the state E2E_SM_INIT.
maxErrorState Invalid	PositiveInteger	0..1	attr	Maximal number of checks in which ProfileStatus equal to E2E_P_ERROR was determined, within the last Window Size checks, for the state E2E_SM_INVALID.

▽

△

Class	EndToEndTransformationDescription			
maxErrorState Valid	PositiveInteger	0..1	attr	Maximal number of checks in which ProfileStatus equal to E2E_P_ERROR was determined, within the last Window Size checks, for the state E2E_SM_VALID.
maxNoNewOr RepeatedData	PositiveInteger	0..1	attr	The maximum allowed amount of consecutive failed counter checks.
minOkStateInit	PositiveInteger	0..1	attr	Minimal number of checks in which ProfileStatus equal to E2E_P_OK was determined, within the last WindowSize checks, for the state E2E_SM_INIT.
minOkState Invalid	PositiveInteger	0..1	attr	Minimal number of checks in which ProfileStatus equal to E2E_P_OK was determined, within the last WindowSize checks, for the state E2E_SM_INVALID.
minOkState Valid	PositiveInteger	0..1	attr	Minimal number of checks in which ProfileStatus equal to E2E_P_OK was determined, within the last WindowSize checks, for the state E2E_SM_VALID.
offset	PositiveInteger	0..1	attr	Offset of the E2E header in the Data[] array in bits.
profileBehavior	EndToEndProfile BehaviorEnum	0..1	attr	Behavior of the check functionality
profileName	NameToken	1	attr	Definition of the E2E profile.
syncCounterInit	PositiveInteger	0..1	attr	Number of checks required for validating the consistency of the counter that shall be received with a valid counter (i.e. counter within the allowed lock-in range) after the detection of an unexpected behavior of a received counter.
upperHeader BitsToShift	PositiveInteger	0..1	attr	This attribute describes the number of upper-header bits to be shifted. value = 0 or not present: shift of upper header is NOT performed. value > 0: the E2E Transformer on the protect-side, takes the first upperHeaderBitsToShift bits from the upper buffer (e.g. SOME/IP header part generated by SOME/IP transformer) and shifts them towards the lower bytes and bits within the Data[] for the length of the E2E header (e.g. 12 bytes in case of E2E Profile 4). This means the shift distance is fixed - it depends on the E2E header size - what is configured here is the number of bits that are to be shifted. This option is defined because the Some/IP header generated by SOME/IP transformer shall be, due to compatibility between non-protected and E2E-protected communication, at the same position, which is before E2E header.
windowSizeInit	PositiveInteger	0..1	attr	Size of the monitoring window of state Init for the E2E state machine.
windowSize Invalid	PositiveInteger	0..1	attr	Size of the monitoring window of state Invalid for the E2E state machine.
windowSize Valid	PositiveInteger	0..1	attr	Size of the monitoring window of state Valid for the E2E state machine.

Table A.470: EndToEndTransformationDescription

Class	<<atpVariation>> **EndToEndTransformationISignalProps**
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer
Note	Holds all the ISignal specific attributes for the EndToEndTransformer.
Base	*ARObject*, *Describable*, *TransformationISignalProps*
Aggregated by	ISignal.transformationISignalProps, ISignalGroup.transformationISignalProps

▽

△

Class	<<atpVariation>> **EndToEndTransformationISignalProps**			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
dataId (ordered)	PositiveInteger	*	attr	This represents a unique numerical identifier.
				Note: ID is used for protection against masquerading. The details concerning the maximum number of values (this information is specific for each E2E profile) applicable for this attribute are controlled by a semantic constraint that depends on the category of the EndToEnd Protection.
dataLength	PositiveInteger	0..1	attr	Length of payload and E2E header in bits.
maxDataLength	PositiveInteger	0..1	attr	Maximum length of payload and E2E header in bits.
minDataLength	PositiveInteger	0..1	attr	Minimum length of payload and E2E header in bits.
sourceId	PositiveInteger	0..1	attr	This attribute represents a unique numerical identifier identifying the source of a certain transmission. In case of C/S communication, this ID uniquely identifies the client.
				Note: ID is used for protection against masquerading. The details concerning the maximum number of values (this information is specific for each E2E profile) applicable for this attribute are controlled by a semantic constraint that depends on the category of the EndToEnd Protection.

Table A.471: EndToEndTransformationISignalProps

Class	**EngineeringObject** (abstract)			
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::EngineeringObject			
Note	This class specifies an engineering object. Usually such an object is represented by a file artifact. The properties of engineering object are such that the artifact can be found by querying an ASAM catalog file.			
	The engineering object is uniquely identified by domain+category+shortLabel+revisionLabel.			
Base	*ARObject*			
Subclasses	AutosarEngineeringObject, BuildEngineeringObject, Graphic			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
category	NameToken	1	attr	This denotes the role of the engineering object in the development cycle. Categories are such as
				• SWSRC for source code
				• SWOBJ for object code
				• SWHDR for a C-header file
				Further roles need to be defined via Methodology.
				Tags:xml.sequenceOffset=20
domain	NameToken	0..1	attr	This denotes the domain in which the engineering object is stored. This allows to indicate various segments in the repository keeping the engineering objects. The domain may segregate companies, as well as automotive domains. Details need to be defined by the Methodology.
				Attribute is optional to support a default domain.
				Tags:xml.sequenceOffset=40
revisionLabel	RevisionLabelString	*	attr	This is a revision label denoting a particular version of the engineering object.
				Tags:xml.sequenceOffset=30

▽

△

Class	EngineeringObject (abstract)			
shortLabel	NameToken	1	attr	This is the short name of the engineering object. Note that it is modeled as NameToken and not as Identifier since in ASAM-CC it is also a NameToken. **Tags:**xml.sequenceOffset=10

Table A.472: EngineeringObject

Class	EnumerationMappingTable			
Package	M2::AUTOSARTemplates::GenericStructure::VariantHandling::AttributeValueVariationPoints			
Note	This class represents an attribute value variation point for Enumeration attributes. Note that this class might be used in the extended meta-model only. **Tags:**atp.recommendedPackage=EnumerationMappingTables			
Base	ARObject, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
entry	EnumerationMapping Entry	*	aggr	Key-value pair mapping enumeration values to unique integers. **Tags:** xml.roleElement=true xml.roleWrapperElement=true xml.typeElement=false xml.typeWrapperElement=false

Table A.473: EnumerationMappingTable

Class	ErrorTracerNeeds			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	Specifies the need to report failures to the error tracer.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable, ServiceNeeds			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	Type	Mult.	Kind	Note
tracedFailure	TracedFailure	*	aggr	list of traced failures **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=tracedFailure.shortName, traced Failure.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table A.474: ErrorTracerNeeds

Class	EthGlobalTimeDomainProps			
Package	M2::AUTOSARTemplates::SystemTemplate::GlobalTime::ETH			
Note	Enables the definition of Ethernet Global Time specific properties.			
Base	ARObject, AbstractGlobalTimeDomainProps			
Aggregated by	GlobalTimeDomain.globalTimeDomainProperty			
Attribute	Type	Mult.	Kind	Note
crcFlags	EthTSynCrcFlags	0..1	aggr	Defines the fields of the message which shall be taken into account for CRC calculation and verification.

▽

△

Class	EthGlobalTimeDomainProps			
destination Physical Address	MacAddressString	0..1	attr	Defines the MAC multicast address the Ethernet time sync messages are communicated on.
fupDataIDList (ordered)	PositiveInteger	0..16	attr	The DataIDList for FUP messages to calculate CRC.
managed CouplingPort	EthGlobalTime ManagedCouplingPort	*	aggr	Collection of CouplingPorts which are managed in the scope of this Ethernet GlobalTimeDomain.
message Compliance	EthGlobalTimeMessage FormatEnum	1	attr	Defines the compliance of the Ethernet time sync messages to specific standards.
vlanPriority	PositiveInteger	0..1	attr	Defines which VLAN priority shall be assigned to a time sync message in case the message is sent using a VLAN tag.

Table A.475: EthGlobalTimeDomainProps

Class	<<atpVariation>> **EthernetCluster**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Ethernet-specific cluster attributes. **Tags:**atp.recommendedPackage=CommunicationClusters			
Base	*ARObject*, *CollectableElement*, *CommunicationCluster*, *FibexElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
couplingPort Connection	CouplingPort Connection	*	aggr	Specification of connections between CouplingElements and EcuInstances. Note: This atpSplitable property has no atp.Splitkey due to atpVariation (PropertySetPattern). **Stereotypes:** atpSplitable; atpVariation **Tags:**vh.latestBindingTime=postBuild
couplingPort StartupActive Time	TimeValue	0..1	attr	The attribute specifies the time in second a coupling port is switched on to enable the host ECU (ECU that maintains an Ethernet switch) to listen to the network for potential network management requests.
couplingPort SwitchoffDelay	TimeValue	0..1	attr	Switch off delay for CouplingPorts in seconds. It denotes the delay of switching off couplingPorts after the request to switch off a couplingPort was issued. (e.g. switch off of Ethernet switch ports).
macMulticast Group	MacMulticastGroup	*	aggr	MacMulticastGroup that is defined for the Subnet (EthernetCluster).

Table A.476: EthernetCluster

Class	EthernetCommunicationConnector			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Ethernet specific attributes to the CommunicationConnector.			
Base	*ARObject*, *CommunicationConnector*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcuInstance.connector, MachineDesign.communicationConnector			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
ethIpProps	EthIpProps	0..1	ref	EcuInstance specific IP attributes.

▽

△

Class	EthernetCommunicationConnector			
maximum Transmission Unit	PositiveInteger	0..1	attr	This attribute specifies the maximum transmission unit in bytes.
neighborCache Size	PositiveInteger	0..1	attr	This attribute specifies the size of neighbor cache or ARP table in units of entries.
pathMtu Enabled	Boolean	0..1	attr	If enabled the IPv4/IPv6 processes incoming ICMP "Packet Too Big" messages and stores a MTU value for each destination address.
pathMtuTimeout	TimeValue	0..1	attr	If this value is >0 the IPv4/IPv6 will reset the MTU value stored for each destination after n seconds.

Table A.477: EthernetCommunicationConnector

Class	<<atpVariation>> **EthernetCommunicationController**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Ethernet specific communication port attributes.			
Base	*ARObject*, *CommunicationController*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcuInstance.commController, MachineDesign.communicationController			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
canXlConfig	AbstractCan Communication Controller	0..1	ref	If the Ethernet frames handled by this Ethernet CommunicationController are to be tunneled through CAN XL, then this reference shall refer to the Abstract CanCommunicationController that aggregates the Can ControllerXlConfiguration of the physical CAN XL channel to be used for tunneling.
couplingPort	CouplingPort	*	aggr	Optional CouplingPort that can be used to connect the ECU to a CouplingElement (e.g. a switch).
macLayerType	EthernetMacLayerType Enum	0..1	attr	Specifies the mac layer type of the ethernet controller.
macUnicast Address	MacAddressString	0..1	attr	Media Access Control address (MAC address) that uniquely identifies each EthernetCommunication Controller in the network.
maximum ReceiveBuffer Length	Integer	0..1	attr	Determines the maximum receive buffer length (frame length) in bytes.
maximum TransmitBuffer Length	Integer	0..1	attr	Determines the maximum transmit buffer length (frame length) in bytes.
slaveActAs Passive Communication Slave	Boolean	0..1	attr	This attribute specifies if the EcuInstance is acting as a passive communication slave on the connected Physical Channel. This is used for EthernetCommunication Controllers that use Ethernet hardware which supports wake-up and sleep on the network (e.g. Open Alliance TC10 compliant Ethernet hardware). **Tags:**atp.Status=draft
slaveQualified UnexpectedLink DownTime	TimeValue	0..1	attr	This attribute specifies time when an unexpected link down is evaluated as link down and indicated to the AUTOSAR communication stack. **Tags:**atp.Status=draft

Table A.478: EthernetCommunicationController

Enumeration	EthernetConnectionNegotiationEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology
Note	Specifies connection negotiation types of Ethernet transceiver links.
Aggregated by	CouplingPort.connectionNegotiationBehavior
Literal	Description
auto	Automatic Negotiation **Tags:**atp.EnumerationLiteralIndex=0
master	Master **Tags:**atp.EnumerationLiteralIndex=1
slave	Slave **Tags:**atp.EnumerationLiteralIndex=2

Table A.479: EthernetConnectionNegotiationEnum

Enumeration	EthernetMacLayerTypeEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology
Note	Specifies MAC (Media Access Control) Layer types.
Aggregated by	CouplingPort.macLayerType, EthernetCommunicationController.macLayerType
Literal	Description
xGMII	Mac layer interface (data) bandwith class 1Gbit/s (e.g. GMII, RGMII, SGMII, RvGMII, USGMII) **Tags:** atp.EnumerationLiteralIndex=1 xml.name=XG-MII
xMII	Mac layer interface (data) bandwith class 100Mbit/s and 10Mbit/s (e.g. RMII, RvMII, SMII, RvMII) **Tags:** atp.EnumerationLiteralIndex=0 xml.name=X-MII
xXGMII	Mac layer interface (data) bandwith class 10Gbit/s **Tags:** atp.EnumerationLiteralIndex=2 xml.name=XXG-MII

Table A.480: EthernetMacLayerTypeEnum

Class	EthernetPhysicalChannel			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	The EthernetPhysicalChannel represents a VLAN or an untagged channel. An untagged channel is modeled as an EthernetPhysicalChannel without an aggregated VLAN.			
Base	ARObject, Identifiable, MultilanguageReferrable, PhysicalChannel, Referrable			
Aggregated by	CommunicationCluster.physicalChannel			
Attribute	Type	Mult.	Kind	Note
network Endpoint	NetworkEndpoint	*	aggr	Collection of NetworkEndpoints that are used in the VLan. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=networkEndpoint.shortName
soAdConfig	SoAdConfig	0..1	aggr	SoAd Configuration for one specific Physical Channel.
vlan	VlanConfig	0..1	aggr	VLAN Configuration.

Table A.481: EthernetPhysicalChannel

Enumeration	EthernetPhysicalLayerTypeEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology
Note	Specifies physical layer types of Ethernet transceiver links.
Aggregated by	CouplingPort.physicalLayerType
Literal	Description
_1000BASE_T	Ethernet Standard (IEEE 802.3ab) to support 1Gbit/s over 4 twisted pairs. **Tags:** atp.EnumerationLiteralIndex=6 xml.name=1000BASE-T
_1000BASE_T1	Ethernet Standard (IEEE 802.3bp) to support 1Gbit/s over a single twisted pair cable. **Tags:** atp.EnumerationLiteralIndex=8 xml.name=1000BASE-T1
_100BASE_T1	Ethernet Standard (IEEE 802.3bw) to support 100Mbit/s over a single twisted pair cable. 100BASE-T1 is the IEEE Standardized version of BroadRReach. **Tags:** atp.EnumerationLiteralIndex=7 xml.name=100BASE-T1
_100BASE_TX	Ethernet Standard (IEEE 802.3u) to support 100Mbit/s over two twisted pairs. **Tags:** atp.EnumerationLiteralIndex=5 xml.name=100BASE-TX
_10BASE_T1S	Physical layer interface 10BASE-T1S (10Mbit/s, 2 pairs). Used for automotive. **Tags:** atp.EnumerationLiteralIndex=10 atp.Status=draft xml.name=10BASE-T1S
iEEE802_11P	Ethernet Standard (IEEE 802.11p) to support wireless communication in vehicular environments. **Tags:** atp.EnumerationLiteralIndex=9 xml.name=IEEE802-11P

Table A.482: EthernetPhysicalLayerTypeEnum

Class	EthernetPriorityRegeneration			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Defines a priority regeneration where the ingressPriority is replaced by regeneratedPriority. The ethernetPriorityRegeneration is optional in case no priority regeneration shall be performed. In case a ethernetPriorityRegeneration is defined it shall have 8 mappings, one for each priority.			
Base	ARObject, Referrable			
Aggregated by	CouplingPortDetails.ethernetPriorityRegeneration			
Attribute	Type	Mult.	Kind	Note
ingressPriority	PositiveInteger	1	attr	Message priority of the incoming message. range: 0-7
regenerated Priority	PositiveInteger	1	attr	Regenerated message priority. range: 0-7

Table A.483: EthernetPriorityRegeneration

Class	EthernetWakeupSleepOnDatalineConfig			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	EthernetWakeupSleepOnDatalineConfigSet is the main element that aggregates different config set regarding the wakeup and sleep on data line. An EthernetWakeupSleepOnDatalineConfigSet could aggregate multiple different configurations regarding the wakeup and sleep on dataline (EthernetWakeupSleepOnDatalineConfig).			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EthernetWakeupSleepOnDatalineConfigSet.ethernetWakeupSleepOnDatalineConfig			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
sleepMode ExecutionDelay	TimeValue	0..1	attr	Delay in seconds to perform a sleep request if the Ethernet hardware (PHY) detect a pending wake-up. This is used to avoid the race condition, if a sleep was requested while a wake-up of a neighboring PHY was received via a local wake-up connection (e.g. I/O pin).
sleepRepetition DelayOfSleep Request	TimeValue	0..1	attr	Delay in seconds for a repetition of a sleep request. This is used to retry a synchronized shutdown of the connected Ethernet hardware (PHY) of the link partner.
sleep RepetitionsOf SleepRequest	PositiveInteger	0..1	attr	Count of repetitions for a sleep on dataline. If a sleep is rejected by the linked communication partner, the sleep is repeated until the count of repetitions exceed. If count of repetitions exceed, the Ethernet hardware (PHY) transit to sleep without acknowledgement of the connected link partner.
wakeupForward LocalEnabled	Boolean	0..1	attr	If enabled, then a remote wake-up received on the physical dataline (e.g. 100BASE-T1) is forwarded as local wake-up (e.g. via an I/O pin). If disabled, then a remote wake-up is not forwarded as local wake-up.
wakeupForward RemoteEnabled	Boolean	0..1	attr	If enabled, then a local wake-up is forwarded to the physical dataline (e.g. 100BASE-T1). If disabled, then a local wake-up is not forwarded to the physical dataline.
wakeupLocal DetectionTime	TimeValue	0..1	attr	Specify the detection time if a local wake-up in seconds is present on the local wake-up connection (e.g. I/O pin). A local wake-up has to be present at least for wakeupLocal DetectionTime to be detected a valid local wake-up.
wakeupLocal DurationTime	TimeValue	0..1	attr	Specify the duration of a local wake-up in seconds to be present on the local wake-up connection (e.g. I/O pin).
wakeupLocal Enabled	Boolean	0..1	attr	If enabled, then a local wake-up received via a local connection (e.g. I/O pin) shall be detected by the Ethernet hardware (PHY). If disabled, Ethernet hardware is not reacting on a local wake-up.
wakeupRemote Enabled	Boolean	0..1	attr	If enabled, then a remote wake-up received via the physical dataline (e.g. 100BASE-T1) shall be detected by the Ethernet hardware (PHY). If disabled, Ethernet hardware is not reaction on a remote wake-up.
wakeup RepetitionDelay OfWakeup Request	TimeValue	0..1	attr	Delay in seconds for a repetition of a wake-up. This is used to increase the reliability in the network, such that an ECU which initiates the wake-up does repeat the wake-up and increase the probability that affected ECUs receive the wake-up.
wakeup RepetitionsOf Wakeup Request	PositiveInteger	0..1	attr	Count of repetitions for a wake-up. This is used to increase the reliability in the network, such that an ECU which initiates the wake-up does repeat the wake-up and increase the probability that affected ECUs receive the wake-up.

Table A.484: EthernetWakeupSleepOnDatalineConfig

Class	EvaluatedVariantSet			
Package	M2::AUTOSARTemplates::GenericStructure::VariantHandling			
Note	This meta class represents the ability to express if a set of ARElements is able to support one or more particular variants. In other words, for a given set of evaluatedElements this meta class represents a table of evaluated variants, where each PredefinedVariant represents one column. In this column each descendant sw SystemconstantValue resp. postbuildVariantCriterionValue represents one entry. In a graphical representation each swSystemconstantValueSet / postBuildVariantCriterionValueSet could be used as an intermediate headline in the table column. If the approvalStatus is "APPROVED" it expresses that the collection of CollectableElements is known be valid for the given evaluatedVariants. Note that the EvaluatedVariantSet is a CollectableElement. This allows to establish a hierarchy of EvaluatedVariantSets. **Tags:**atp.recommendedPackage=EvaluatedVariantSets			
Base	*ARElement*, ARObject, CollectableElement, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
approvalStatus	NameToken	1	attr	Defines the approval status of a predefined variant. Two values are predefined: "APPROVED" and "REJECTED": • Approved variants are known to work. • Rejected variants are known NOT to work. Further values can be approved on a per-company basis; within AUTOSAR only "APPROVED" and "REJECTED" should be recognized.
evaluated Element	CollectableElement	*	ref	This represents a particular element which is evaluated in context of the EvaluatedVariants. The approvalStatus applies to this element (and all of its descendants). In other words, the referenced elements are those that were considered when the predefined variant was evaluated.
evaluated Variant	PredefinedVariant	*	ref	This metaclass represents one particular variant which was evaluated. LowerMultiplicity is set to 0 to support a stepwise approach.

Table A.485: EvaluatedVariantSet

Enumeration	EventGroupControlTypeEnum	
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances	
Note	Types of a RoutingGroups for the event communication.	
Aggregated by	PduActivationRoutingGroup.eventGroupControlType, SoAdRoutingGroup.eventGroupControlType	
Literal	Description	
activationAnd TriggerUnicast	Activate the data path for unicast events and triggered unicast events that are sent out after a client got subscribed. **Tags:**atp.EnumerationLiteralIndex=0	
activationMulticast	Activate the data path for multicast events of an EventGroup. **Tags:**atp.EnumerationLiteralIndex=1	
activationUnicast	Activate the data path for unicast events of an EventGroup. **Tags:**atp.EnumerationLiteralIndex=2	
triggerUnicast	Activate the data path for triggered unicast events that are sent out after a client got subscribed. **Tags:**atp.EnumerationLiteralIndex=3	

Table A.486: EventGroupControlTypeEnum

Class	EventHandler			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances			
Note	This element represents an event group as part of the Provided Service Instance.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ProvidedServiceInstance.eventHandler			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
consumedEvent Group	ConsumedEventGroup	*	ref	All consumers of the event are referenced here. **Tags:**atp.Status=obsolete
eventGroup Identifier	PositiveInteger	0..1	attr	Unique Identifier that identifies the EventGroup in SOME/IP. This Identifier is sent as Eventgroup ID in SOME/IP Service Discovery messages.
eventMulticast Address	ApplicationEndpoint	0..1	ref	Multicast Address that is used for event communication in the IP-Multicast case. It is the destination address to which the server sends the multicast event messages if the mulicastThreshold is exceeded. This address is transmitted in the SD-SubscribeEvent GroupAck Message to client (answer to SD-Subscribe EventGroup). **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=eventMulticastAddress.applicationEndpoint, eventMulticastAddress.variationPoint.shortLabel vh.latestBindingTime=postBuild
multicast Threshold	PositiveInteger	0..1	attr	Specifies the number of subscribed clients that trigger the server to change the transmission of events to multicast. If configured to 0 only unicast will be used. If configured to 1 the first client will be already served by multicast. If configured to 2 the first client will be server with unicast and as soon as the second client arrives both will be served by multicast. This does not influence the handling of initial events, which are served using unicast only.
pduActivation RoutingGroup	PduActivationRouting Group	*	aggr	The ServiceDiscovery module is able to activate and deactivate the PDU routing for events.
routingGroup	SoAdRoutingGroup	*	ref	The ServiceDiscovery module is able to activate and deactivate the PDU routing for events. **Tags:**atp.Status=obsolete
sdServerConfig	SdServerConfig	0..1	aggr	Server configuration parameter for Service-Discovery. **Tags:**atp.Status=obsolete
sdServerEg TimingConfig	SomeipSdServerEvent GroupTimingConfig	0..1	ref	Server Timing configuration settings that are EventGroup specific. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=sdServerEgTimingConfig.someipSdServer EventGroupTimingConfig, sdServerEgTiming Config.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.487: EventHandler

Class	EventTriggeringConstraint (abstract)			
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint::EventTriggeringConstraint			
Note	Describes the occurrence behavior of the referenced timing event. 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.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable, TimingConstraint, Traceable			
Subclasses	ArbitraryEventTriggering, BurstPatternEventTriggering, ConcretePatternEventTriggering, PeriodicEvent Triggering, SporadicEventTriggering			
Aggregated by	TimingExtension.timingGuarantee, TimingExtension.timingRequirement			
Attribute	Type	Mult.	Kind	Note
event	TimingDescriptionEvent	0..1	ref	The referenced timing event

Table A.488: EventTriggeringConstraint

Class	ExclusiveArea			
Package	M2::AUTOSARTemplates::CommonStructure::InternalBehavior			
Note	Prevents an executable entity running in the area from being preempted.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	InternalBehavior.exclusiveArea			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.489: ExclusiveArea

Class	ExecutableEntity (abstract)			
Package	M2::AUTOSARTemplates::CommonStructure::InternalBehavior			
Note	Abstraction of executable code.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Subclasses	BswModuleEntity, RunnableEntity			
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.
canEnter	ExclusiveArea	*	ref	This means that the executable entity can enter/leave the referenced exclusive area through explicit API calls. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=canEnter.exclusiveArea, canEnter.variation Point.shortLabel vh.latestBindingTime=preCompileTime
exclusiveArea NestingOrder	ExclusiveAreaNesting Order	*	ref	This represents the set of ExclusiveAreaNestingOrders recognized by this ExecutableEntity.
minimumStart Interval	TimeValue	0..1	attr	Specifies the time in seconds by which two consecutive starts of an ExecutableEntity are guaranteed to be separated.

▽

△

Class	ExecutableEntity (abstract)			
reentrancyLevel	ReentrancyLevelEnum	0..1	attr	The reentrancy level of this ExecutableEntity. See the documentation of the enumeration type ReentrancyLevel Enum for details.

Please note that nonReentrant interfaces can have also reentrant or multicoreReentrant implementations, and reentrant interfaces can also have multicoreReentrant implementations. |
| runsInside | ExclusiveArea | * | ref | The executable entity runs completely inside the referenced exclusive area.

Stereotypes: atpSplitable; atpVariation
Tags:
atp.Splitkey=runsInside.exclusiveArea, runs Inside.variationPoint.shortLabel
vh.latestBindingTime=preCompileTime |
| swAddrMethod | SwAddrMethod | 0..1 | ref | 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. |

Table A.490: ExecutableEntity

Class	ExecutableEntityActivationReason			
Package	M2::AUTOSARTemplates::CommonStructure::InternalBehavior			
Note	This meta-class represents the ability to define the reason for the activation of the enclosing Executable Entity.			
Base	ARObject, ImplementationProps, Referrable			
Aggregated by	ExecutableEntity.activationReason			
Attribute	Type	Mult.	Kind	Note
bitPosition	PositiveInteger	0..1	attr	This attribute allows for defining the position of the enclosing ExecutableEntityActivationReason in the activation vector.

Table A.491: ExecutableEntityActivationReason

Class	ExecutionOrderConstraint				
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint::ExecutionOrderConstraint				
Note	This constraint is used to restrict the order of execution for a set of ExecutableEntities. The Execution OrderConstraint can be used in any timing view. The various scopes for ExecutionOrderConstraint are described below. Generally, each ExecutionOrder Constraint has a scope of software components and can reference all executable entities available in the corresponding internal behavior (RunnableEntity and BswModuleEntity) either directly or by the events activating respectively starting them (RteEvent and BswEvent). On VFB level an ExecutionOrderConstraint can be specified for RunnableEntities part of the composition hierarchy referenced by the VfbTiming. The ExecutionOrderConstraint is aggregated by the VfbTiming. On SW-C level an ExecutionOrderConstraint can be specified for RunnableEntities part of the Internal Behavior referenced by the SwcTiming. The ExecutionOrderConstraint is aggregated by the SwcTiming. On System level an ExecutionOrderConstraint can be specified for RunnableEntities part of the composition hierarchy of the system referenced by the SystemTiming. The ExecutionOrderConstraint is aggregated by the SystemTiming. On BSW Module level, an ExectionOrderConstraint can be specified for BswModuleEntities part of an BswInternalBehavior referenced by the BswModuleTiming. The ExecutionOrderConstraint is aggregated by the BswModuleTiming. On ECU level an ExecutionOrderConstraint can be specified for all ExecutableEntities and Events available via the EcucValueCollection, covering ECU Extract and BSW Module Configuration, referenced by the EcuTiming. The ExecutionOrderConstraint is aggregated by the EcuTiming.				
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable, TimingConstraint, Traceable				
Aggregated by	TimingExtension.timingGuarantee, TimingExtension.timingRequirement				
Attribute	Type	Mult.	Kind	Note	
baseComposition	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.	
executionOrderConstraintType	ExecutionOrderConstraintTypeEnum	0..1	attr	Specifies the specific type of ExecutionOrderConstraint.	
ignoreOrderAllowed	Boolean	0..1	attr	Controls whether the order of execution specified by this constraint can be intentionally ignored (TRUE), or shall be respected (FALSE).	
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 A.492: ExecutionOrderConstraint

Class	ExecutionTime (abstract)				
Package	M2::AUTOSARTemplates::CommonStructure::ResourceConsumption::ExecutionTime				
Note	Base class for several means how to describe the ExecutionTime of software. The required context information is provided through this class.				
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable				
Subclasses	AnalyzedExecutionTime, MeasuredExecutionTime, RoughEstimateOfExecutionTime, SimulatedExecutionTime				
Aggregated by	ResourceConsumption.executionTime				
Attribute	Type	Mult.	Kind	Note	

△

Class	ExecutionTime (abstract)			
exclusiveArea	ExclusiveArea	0..1	ref	Reference to the ExclusiveArea this execution time is provided for.
executableEntity	ExecutableEntity	0..1	ref	The executable entity for which this execution time is described.
hardware Configuration	HardwareConfiguration	0..1	aggr	Provides information on the HardwareConfiguration used to specify this ExecutionTime.
hwElement	HwElement	0..1	ref	The hardware element (e.g. type of ECU) for which the execution time is specified.
includedLibrary	DependencyOnArtifact	*	ref	If this dependency is specified, the execution time of the library code is included in the execution time data for the runnable.
memorySection Location	MemorySectionLocation	*	aggr	Provides information on the MemorySectionLocation which is involved in the ExecutionTime description.
softwareContext	SoftwareContext	0..1	aggr	Provides information on the detailed SoftwareContext used to provide the ExecutionTime description.

Table A.493: ExecutionTime

Class	ExecutionTimeConstraint			
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint::ExecutionTimeConstraint			
Note	An ExecutionTimeConstraint is used to specify the execution time of the referenced ExecutableEntity in the referenced component. A minimum and maximum execution time can be defined. Two types of execution time semantics can be used. The desired semantics can be set by the attribute executionTimeType: • The "net" execution time is the time used to execute the ExecutableEntity without interruption and without external calls. • The "gross" execution time is the time used to execute the ExecutableEntity without interruption including external calls to other entities. The time to execute the ExecutableEntity 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.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable, TimingConstraint, Traceable			
Aggregated by	TimingExtension.timingGuarantee, TimingExtension.timingRequirement			
Attribute	Type	Mult.	Kind	Note
component	SwComponent Prototype	0..1	iref	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. **InstanceRef implemented by:**ComponentInComposition InstanceRef
executable	ExecutableEntity	0..1	ref	The referenced ExecutableEntity for the ExecutionTime Constraint.
executionTime Type	ExecutionTimeType Enum	0..1	attr	Specifies the type of the execution time constrained by ExecutionTimeConstraint,
maximum	MultidimensionalTime	0..1	aggr	The maximum execution time.
minimum	MultidimensionalTime	0..1	aggr	The minimum execution time.

Table A.494: ExecutionTimeConstraint

Class	ExternalTriggerOccurredEvent			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents			
Note	This event is raised when the referenced Trigger has occurred.			
Base	ARObject, AbstractEvent, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable, Multilanguage Referrable, RTEEvent, Referrable			
Aggregated by	AtpClassifier.atpFeature, SwcInternalBehavior.event			
Attribute	Type	Mult.	Kind	Note
trigger	Trigger	0..1	iref	The referenced Trigger raises this ExternalTrigger OccurredEvent. **InstanceRef implemented by:**RTriggerInAtomicSwc InstanceRef

Table A.495: ExternalTriggerOccurredEvent

Class	ExternalTriggeringPoint			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::Trigger			
Note	If a RunnableEntity owns an ExternalTriggeringPoint it is entitled to raise an ExternalTriggerOccurred Event.			
Base	ARObject			
Aggregated by	RunnableEntity.externalTriggeringPoint			
Attribute	Type	Mult.	Kind	Note
ident	ExternalTriggeringPoint Ident	0..1	aggr	The aggregation in the role ident provides the ability to make the ExternalTriggeringPoint identifiable. From the semantical point of view, the ExternalTriggering Point is considered a first-class Identifiable and therefore the aggregation in the role ident shall always exist (until it may be possible to let ModeAccessPoint directly inherit from Identifiable). **Stereotypes:** atpIdentityContributor **Tags:**xml.sequenceOffset=-100
trigger	Trigger	0..1	iref	The trigger taken for the ExternalTriggeringPoint. **Tags:** xml.namePlural=TRIGGER-IREF xml.roleElement=false xml.roleWrapperElement=true xml.typeElement=true xml.typeWrapperElement=false **InstanceRef implemented by:**PTriggerInAtomicSwcType InstanceRef

Table A.496: ExternalTriggeringPoint

Class	FibexElement (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore			
Note	ASAM FIBEX elements specifying Communication and Topology.			
Base	ARObject, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Subclasses	BusMirrorChannelMapping, CommunicationCluster, ConsumedProvidedServiceInstanceGroup, CouplingElement, EcuInstance, EthernetWakeupSleepOnDatalineConfigSet, Frame, Gateway, Global TimeDomain, ISignal, ISignalGroup, ISignalIPduGroup, NmConfig, Pdu, PdurIPduGroup, Secure CommunicationPropsSet, ServiceInstanceCollectionSet, SoAdRoutingGroup, SocketConnectionIpdu IdentifierSet, TpConfig			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.497: FibexElement

Class	FlatInstanceDescriptor			
Package	M2::AUTOSARTemplates::CommonStructure::FlatMap			
Note	Represents exactly one node (e.g. a component instance or data element) of the instance tree of a software system. The purpose of this element is to map the various nested representations of this instance to a flat representation and assign a unique name (shortName) to it. Use cases: • Specify unique names of measurable data to be used by MCD tools • Specify unique names of calibration data to be used by MCD tool • Specify a unique name for an instance of a component prototype in the ECU extract of the system description Note that in addition it is possible to assign alias names via AliasNameAssignment.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	FlatMap.instance			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
ecuExtract Reference	AtpFeature	0..1	iref	Refers to the instance in the ECU extract. This is valid only, if the FlatMap is used in the context of an ECU extract. The reference shall be such that it uniquely defines the object instance. For example, if a data prototype is declared as a role within an SwcInternalBehavior, it is not enough to state the SwcInternalBehavior as context and the aggregated data prototype as target. In addition, the reference shall also include the complete path identifying instance of the component prototype and the Atomic SoftwareComponentType, which is refered by the particular SwcInternalBehavior. **Tags:**xml.sequenceOffset=40 **InstanceRef implemented by:**AnyInstanceRef
role	Identifier	0..1	attr	The role denotes the particular role of the downstream memory location described by this FlatInstanceDescriptor. It applies to use case where one upstream object results in multiple downstream objects, e.g. ModeDeclaration GroupPrototypes which are measurable. In this case the RTE will provide locations for current mode, previous mode and next mode.
rtePluginProps	RtePluginProps	0..1	aggr	The properties of a communication graph with respect to the utilization of RTE Implementation Plug-in. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=rtePluginProps
swDataDef Props	SwDataDefProps	0..1	aggr	The properties of this FlatInstanceDescriptor. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=swDataDefProps

▽

△

Class	FlatInstanceDescriptor			
upstream Reference	AtpFeature	0..1	iref	Refers to the instance in the context of an "upstream" descriptions, wich could be the system or system extract description, the basic software module description or (if a flat map is used in preliminary context) a description of an atomic component or composition. This reference is optional in case the flat map is used in ECU context. The reference shall be such that it uniquely defines the object instance in the given context. For example, if a data prototype is declared as a role within an SwcInternal Behavior, it is not enough to state the SwcInternal Behavior as context and the aggregated data prototype as target. In addition, the reference shall also include the complete path identifying the instance of the component prototype that contains the particular instance of Swc InternalBehavior. **Tags:**xml.sequenceOffset=20 **InstanceRef implemented by:**AnyInstanceRef

Table A.498: FlatInstanceDescriptor

Class	FlexrayArTpChannel			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	A channel is a group of connections sharing several properties. The FlexRay AutosarTransport Layer supports several channels. These channels can work concurrently, thus each of them requires its own state machine and management data structures and its own PDU-IDs.			
Base	ARObject			
Aggregated by	FlexrayArTpConfig.tpChannel			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
ackType	FrArTpAckType	1	attr	Type of Acknowledgement.
cancellation	Boolean	0..1	attr	With this switch Tx and Rx Cancellation can be turned on or off.
extended Addressing	Boolean	1	attr	Adressing Type of this connection: true: Two Bytes false: One Byte
maxAr	Integer	0..1	attr	This attribute defines the maximum number of trying to send a frame when a TIMEOUT AR occurs (depending on whether retry is configured).
maxAs	Integer	0..1	attr	This attribute defines the maximum number of trying to send a frame when a TIMEOUT AS occurs (depending on whether retry is configured).
maxBs	Integer	0..1	attr	This attribute defines the number of consecutive CFs between two FCs (block size). Valid values are 1 .. 16 when retry is activated, and 0 .. 255 otherwise.
maxFcWait	PositiveInteger	0..1	attr	This attribute defines the maximal number of wait frames to be sent for a pending connection. Range is 0..255.
maximum MessageLength	MaximumMessage LengthType	1	attr	This specifies the maximum message length for the particular channel.
maxRetries	Integer	0..1	attr	This attribute defines the maximum number of retries (if retry is configured for the particular channel).

▽

Class	**FlexrayArTpChannel**			
minimum Multicast SeperationTime	TimeValue	0..1	attr	This attribute defines the minimum amount of time between two succeeding CFs of a 1:n segmented transmission in seconds. Valid values are 0, 100μs, 200μs ... 900μs, 1ms, 2ms .. 127ms. The value can be changed at runtime using the FrArTp_ChangeParameter interface.
				minimumMulticastSeparationTime shall be an integer multiple of the cycle length multiplied with the multiplexing factor, i.e. minimumMulticastSeparationTime = n * cycle * m, where n is an integer >= 0, cycle is Flexray Cluster.cycle, and m is the cycle multiplexor of those cycles where PDUs of the PDU pool are scheduled. Please note: Due to the scheduling strategies of FrTp, minimumMulticastSeparationTime can only be kept to a degree defined by the maximum temporal distance of the PDUs of a PDU pool within one FlexRay cycle.
				Range: 0 .. 0.127
minimum SeparationTime	TimeValue	1	attr	This attribute defines the minimum amount of time between two succeeding CFs of a 1:1 segmented transmission in seconds. Valid values are 0, 100μs, 200μs .. 900μs, 1ms, 2ms .. 127ms. The value can be changed at runtime using the FrArTp_ChangeParameter interface.
				The minimumSeparationTime shall be an integer multiple of the cycle length multiplied with the multiplexing factor, i.e. minimumSeparationTime = n * cycle * m, where n is an integer >=0, cycle is FlexrayCluster.cycle, and m is the cycle multiplexor of those cycles where PDUs of the PDU pool are scheduled.
				Please note: Due to the scheduling strategies of FrTp, minimumSeparationTime can only be kept to a degree defined by the maximum temporal distance of the PDUs of a PDU pool within one FlexRay cycle.
				Range: 0 .. 0.127
multicast Segmentation	Boolean	1	attr	This attribute defines whether segmentation within a 1:n connection is allowed or not.
nPdu	NPdu	*	ref	A FlexRayTpChannel references a set of NPdus. These NPdus are logically assembled into a pool of Rx NPdus and another pool of Tx NPdus. It shall be ensured that a second channel either references all NPdus of such a pool, or none.
timeBr	TimeValue	0..1	attr	This attribute defines the time in seconds between receiving the last CF of a block or an FF-x (or SF-x) and sending out an FC or AF.
timeCs	TimeValue	0..1	attr	This attribute defines the time in seconds between the sending of two consecutive frames or between a consecutive frame and a flow control (for Transmit Cancellation) or between reception of an flow control or Acknowledgement Frame and sending of the next consecutive frame or a flow control (for Transmit Cancellation).
timeoutAr	TimeValue	0..1	attr	This attribute states the timeout in seconds between the PDU transmit request of the Transport Layer to the Flex Ray Interface and the corresponding confirmation of the FlexRay Interface on the receiver side (for FC or AF).

△

Class	FlexrayArTpChannel			
timeoutAs	TimeValue	0..1	attr	This attribute states the timeout in seconds between the PDU transmit request for the first PDU of the group used in the current connection of the Transport Layer to the FlexRay Interface and the corresponding confirmation of the FlexRay Interface (when having sent the last PDU of the group used in this connection) on the sender side (SF-x, FF-x, CF).
timeoutBs	TimeValue	0..1	attr	This attribute defines the timeout in seconds for waiting for an FC or AF on the sender side in a 1:1 connection.
timeoutCr	TimeValue	0..1	attr	This attribute defines the timeout value in seconds for waiting for a CF or FF-x (in case of retry) after receiving the last CF or after sending an FC or AF on the receiver side.
tpConnection	FlexrayArTpConnection	1..*	aggr	Group of connections that can be used in this channel.

Table A.499: FlexrayArTpChannel

Class	FlexrayArTpConnection			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	A connection within a channel identifies the sender and the receiver of this particular communication. The FlexRay Autosar Tp module routes a Pdu through this connection.			
Base	ARObject, TpConnection			
Aggregated by	FlexrayArTpChannel.tpConnection			
Attribute	Type	Mult.	Kind	Note
connectionPrio Pdus	Integer	0..1	attr	This parameter defines the number of PDUs that shall be reserved for this connection when it is active. The range is 1-255.
directTpSdu	IPdu	1	ref	Reference to the IPdu that is segmented by the Transport Protocol. The source address of the transmitted NPdu is determined by the configured source Communication Connector. The target address of the transmitted NPdu is determined by the configured target Communication Connector.
multicast	TpAddress	0..1	ref	TP address for 1:n connections.
reversedTpSdu	IPdu	0..1	ref	Reference to the IPdu that is segmented by the Transport Protocol. If support of both sending and receiving is used, this association references the IPdu used for the additional second direction. The source address of the transmitted NPdu is determined by the configured target Communication Connector. The target address of the transmitted NPdu is determined by the configured source Communication Connector.
source	FlexrayArTpNode	1	ref	The source of the TP connection.
target	FlexrayArTpNode	1..*	ref	The target of the TP connection.

Table A.500: FlexrayArTpConnection

Class	FlexrayArTpNode			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	TP Node (Sender or Receiver) provides the TP Address and the connection to the Topology description.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	FlexrayArTpConfig.tpNode			
Attribute	Type	Mult.	Kind	Note
connector	FlexrayCommunication Connector	*	ref	Association to one or more physical connectors (max number of connectors for FlexRay: 2). In a System Description this reference is mandatory. In an ECU Extract this reference is optional (references to ECUs that are not part of the ECU Extract shall be avoided).
tpAddress	TpAddress	0..1	ref	Reference to the TP Address that is used by the TpNode. This reference is optional in case that the multicast TP Address is used (reference from TpConnection).

Table A.501: FlexrayArTpNode

Enumeration	FlexrayChannelName
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology
Note	Name of the channel.
Aggregated by	FlexrayPhysicalChannel.channelName
Literal	Description
channelA	Channel A **Tags:**atp.EnumerationLiteralIndex=0
channelB	Channel B **Tags:**atp.EnumerationLiteralIndex=1

Table A.502: FlexrayChannelName

Class	<<atpVariation>> FlexrayCluster			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology			
Note	FlexRay specific attributes to the physicalCluster **Tags:**atp.recommendedPackage=CommunicationClusters			
Base	*ARObject*, *CollectableElement*, *CommunicationCluster*, *FibexElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
actionPoint Offset	Integer	1	attr	The offset of the action point in networks
bit	TimeValue	1	attr	Nominal bit time (= 1 / fx:SPEED). gdBit = cSamplesPer Bit * gdSampleClockPeriod. Unit: seconds (gdBit)
casRxLowMax	Integer	1	attr	Upper limit of the Collision Avoidance Symbol (CAS) acceptance window. Unit:bitDuration
coldStart Attempts	Integer	1	attr	The maximum number of times that a node in this cluster is permitted to attempt to start the cluster by initiating schedule synchronization
cycle	TimeValue	1	attr	Length of the cycle. Unit: seconds
cycleCountMax	Integer	1	attr	Maximum cycle counter value in a given cluster. Remark: Set to 63 for FlexRay Protocol 2.1 Rev. A compliance.

$\bigtriangledown$

△

Class	<<atpVariation>> **FlexrayCluster**			
detectNitError	Boolean	1	attr	Indicates whether NIT error status of each cluster shall be detected or not.
dynamicSlotIdle Phase	Integer	1	attr	The duration of the dynamic slot idle phase in minislots.
ignoreAfterTx	Integer	1	attr	Duration for which the bitstrobing is paused after transmission [gdBit].
listenNoise	Integer	1	attr	Upper limit for the start up and wake up listen timeout in the presence of noise. Expressed as a multiple of the cluster constant pdListenTimeout. Unit microticks
macroPerCycle	Integer	1	attr	The number of macroticks in a communication cycle
macrotick Duration	TimeValue	1	attr	Duration of the cluster wide nominal macrotick, expressed in s.
maxWithout ClockCorrection Fatal	Integer	1	attr	Threshold concerning vClockCorrectionFailedCounter. Defines the number of consecutive even/odd Cycle pairs with missing clock correction terms that will cause the protocol to transition from the POC:normal active or POC:normal passive state into the POC:halt state.
maxWithout ClockCorrection Passive	Integer	1	attr	Threshold concerning vClockCorrectionFailedCounter. Defines the number of consecutive even/odd Cycle pairs with missing clock correction terms that will cause the protocol to transition from the POC:normal active state to the POC:normal passive state.
minislotAction PointOffset	Integer	1	attr	The Offset of the action point within a minislot. Unit: macroticks
minislotDuration	Integer	1	attr	The duration of a minislot (dynamic segment). Unit: macroticks.
networkIdle Time	Integer	1	attr	The duration of the network idle time in macroticks
network Management VectorLength	Integer	1	attr	Length of the Network Management vector in a cluster [bytes]
numberOf Minislots	Integer	1	attr	Number of Minislots in the dynamic segment.
numberOfStatic Slots	Integer	1	attr	The number of static slots in the static segment.
offsetCorrection Start	Integer	1	attr	Start of the offset correction phase within the Network Idle Time (NIT), expressed as the number of macroticks from the start of cycle. Unit: macroticks
payloadLength Static	Integer	1	attr	Globally configured payload length of a static frame. Unit: 16-bit WORDS.
safetyMargin	Integer	1	attr	Additional timespan in macroticks which takes jitter into account to be able to set the JobListPointer to the next possible job which can be executed in case the FlexRay Job List Execution Function has be resynchronized.
sampleClock Period	TimeValue	0..1	attr	Sample clock period. Unit: seconds
staticSlot Duration	Integer	1	attr	The duration of a slot in the static segment. Unit: macroticks
symbolWindow	Integer	1	attr	The duration of the symbol window. Unit: macroticks
symbolWindow ActionPoint Offset	Integer	1	attr	Number of macroticks the action point offset is from the beginning of the symbol window [Macroticks].
syncFrameId CountMax	Integer	1	attr	Maximum number of distinct syncframe identifiers present in a given cluster. This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gSyncNodeMax.

▽

△

Class	<<atpVariation>> FlexrayCluster			
tranceiver StandbyDelay	Float	0..1	attr	The duration of timer t_TrcvStdbyDelay in seconds. The granularity of this parameter shall be restricted to full Flex Ray cycles (cycle). The transceiver status setting to STANDBY shall be delayed by this value. Not specifying a value or a value of 0 shall imply that the timer is not used.
transmission StartSequence Duration	Integer	1	attr	Number of bits in the Transmission Start Sequence [gd Bits].
wakeupRxIdle	Integer	1	attr	Number of bits used by the node to test the duration of the 'idle' or HIGH phase of a received wakeup. Unit:bit Duration Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolRxIdle.
wakeupRxLow	Integer	1	attr	Number of bits used by the node to test the duration of the LOW phase of a received wakeup. Unit:bitDuration Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolRxLow.
wakeupRx Window	Integer	1	attr	The size of the window used to detect wakeups [gdBit]. Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolRxWindow.
wakeupTxActive	Integer	1	attr	Number of bits used by the node to transmit the LOW phase of awakeup symbol and the HIGH and LOW phases of a WUDOP. Unit:bitDuration
wakeupTxIdle	Integer	1	attr	Number of bits used by the node to transmit the 'idle' part of a wakeup symbol. Unit: gDbit

Table A.503: FlexrayCluster

Class	FlexrayCommunicationConnector			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology			
Note	FlexRay specific attributes to the CommunicationConnector			
Base	*ARObject*, *CommunicationConnector*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcuInstance.connector, MachineDesign.communicationConnector			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
nmReadySleep Time	Float	0..1	attr	The value of this attribute influences the shutdown behavior of the FlexRay NM. FrNm switches to bus sleep mode nmReadySleepTime seconds after the completion of the last repetition cycle containing a NM vote.
wakeUp Channel	Boolean	1	attr	Referenced channel used by the node to send a wakeup pattern. (pWakeupChannel)

Table A.504: FlexrayCommunicationConnector

Class	FlexrayFrame
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Flexray::FlexrayCommunication
Note	FlexRay specific Frame element. **Tags:**atp.recommendedPackage=Frames
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Frame*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*

▽

△

Class	FlexrayFrame			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.505: FlexrayFrame

Class	FlexrayFrameTriggering			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Flexray::FlexrayCommunication			
Note	FlexRay specific attributes to the FrameTriggering			
Base	ARObject, *FrameTriggering*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*PhysicalChannel*.frameTriggering			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
absolutely Scheduled Timing	FlexrayAbsolutely ScheduledTiming	*	aggr	Specification of a sending behaviour where the exact time for the frames transmission is guaranteed.
allowDynamic LSduLength	Boolean	1	attr	Allows L-PDU length reduction and indicates that the related CC buffer has to be reconfigured for the actual length and Header-CRC before transmission of the L-PDU. If this attribute is set to true than the referenced Frame length attribute defines the max. length.
messageId	PositiveInteger	0..1	attr	The first two bytes of the payload segment of the FlexRay frame format for frames transmitted in the dynamic segment can be used as receiver filterable data called the message ID.
payload Preamble Indicator	Boolean	1	attr	Switching the Payload Preamble bit.

Table A.506: FlexrayFrameTriggering

Class	FlexrayNmCluster			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	FlexRay specific NM cluster attributes.			
Base	ARObject, *Identifiable*, *MultilanguageReferrable*, *NmCluster*, *Referrable*			
Aggregated by	NmConfig.nmCluster			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
nmCarWakeUp BitPosition	PositiveInteger	0..1	attr	Specifies the bit position of the CarWakeUp within the Nm Pdu.
nmCarWakeUp FilterEnabled	Boolean	0..1	attr	If this attribute is set to true the CareWakeUp filtering is supported. In this case only the CarWakeUp bit within the NmPdu with source node identifier nmCarWakeUpFilter NodeId is considered as CarWakeUp request.
nmCarWakeUp FilterNodeId	PositiveInteger	0..1	attr	Source node identifier for CarWakeUp filtering. If Car WakeUp filtering is supported (nmCarWakeUpFilter Enabled), only the CarWakeUp bit within the NmPdu with source node identifier nmCarWakeUpFilterNodeId is considered as CarWakeUp request.
nmCarWakeUp RxEnabled	Boolean	0..1	attr	If set to true this attribute enables the support of CarWake Up bit evaluation in received NmPdus.

▽

△

Class	FlexrayNmCluster				
nmDataCycle	Integer	1	attr	Number of FlexRay Communication Cycles needed to transmit the Nm Data PDUs of all FlexRay Nm Ecus of this FlexRayNmCluster.	
nmMain FunctionPeriod	TimeValue	0..1	attr	Defines the processing cycle of the main function of FrNm module.	
nmRemote SleepIndication Time	TimeValue	1	attr	Timeout for Remote Sleep Indication in seconds. It defines the time how long it shall take to recognize that all other nodes are ready to sleep.	
nmRepeat MessageTime	TimeValue	1	attr	Timeout for Repeat Message State in seconds. Defines the time how long the NM shall stay in the Repeat Message State.	
nmRepetition Cycle	Integer	1	attr	Number of FlexRay Communication Cycles used to repeat the transmission of the Nm vote Pdus of all Flex Ray NmEcus of this FlexRayNmCluster. This value shall be an integral multiple of nmVotingCycle.	
nmVotingCycle	Integer	1	attr	Number of FlexRay CommunicationCycles needed to transmit the Nm vote of Pdus of all FlexRay NmEcus of this FlexRayNmCluster.	

Table A.507: FlexrayNmCluster

Class	FlexrayPhysicalChannel				
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology				
Note	FlexRay specific attributes to the physicalChannel				
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *PhysicalChannel*, *Referrable*				
Aggregated by	*CommunicationCluster*.physicalChannel				
Attribute	Type	Mult.	Kind	Note	
channelName	FlexrayChannelName	1	attr	Name of the channel (Channel A or Channel B).	

Table A.508: FlexrayPhysicalChannel

Class	FlexrayTpConnection				
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols				
Note	A connection identifies the sender and the receiver of this particular communication. The FlexRayTp module routes a Pdu through this connection. In a System Description the references to the PduPools are mandatory. In an ECU Extract these references can be optional: On unicast connections these references are always mandatory. On multicast the txPduPool is mandatory on the sender side. The rxPduPool is mandatory on the receiver side. On Gateway ECUs both references are mandatory.				
Base	*ARObject*, *TpConnection*				
Aggregated by	FlexrayTpConfig.tpConnection				
Attribute	Type	Mult.	Kind	Note	
bandwidth Limitation	Boolean	1	attr	Specifies whether the connection requires a bandwidth limitation or not.	
directTpSdu	IPdu	1	ref	Reference to the IPdu that is segmented by the Transport Protocol.	
multicast	TpAddress	0..1	ref	TP address for 1:n connections.	
receiver	FlexrayTpNode	1..*	ref	The target of the TP connection.	

▽

△

Class	**FlexrayTpConnection**			
reversedTpSdu	IPdu	0..1	ref	Reference to the IPdu that is segmented by the Transport Protocol. If support of both sending and receiving is used, this association references the IPdu used for the additional second direction.
rxPduPool	FlexrayTpPduPool	0..1	ref	A connection has a reference to a set of NPdus (FrTpRx PduPool) which are defined for receiving data via this particular connection. The following constraint is valid only for the System Extract/ECU Extract: In case this connection is applied to the transmitter the rxPduPool holds the actually received NPdus. In case this connection is applied to the receiver the rxPduPool holds the actually sent NPdus.
tpConnection Control	FlexrayTpConnection Control	1	ref	Reference to the connection control.
transmitter	FlexrayTpNode	1	ref	The source of the TP connection.
txPduPool	FlexrayTpPduPool	0..1	ref	A connection has a reference to a set of NPdus (FrTpTx PduPool) which are defined for sending data via this particular connection. The following constraint is valid only for the System Extract/ECU Extract: In case this connection is applied to the transmitter the txPduPool holds the actually sent NPdus. In case this connection is applied to the receiver the txPduPool holds the actually received NPdus.

Table A.509: FlexrayTpConnection

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	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). The frameLength of zero bytes is allowed. Please consider also TPS_SYST_02255.
pduToFrame Mapping	PduToFrameMapping	*	aggr	A frames layout as a sequence of Pdus. atpVariation: The content of a frame can be variable. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=pduToFrameMapping.shortName, pduTo FrameMapping.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.510: Frame

Class	FramePid			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	Frame_PIDs that are included in the request. The "pid" attribute describes the value and the "index" attribute the position of the frame_PID in the request.			
Base	ARObject			
Aggregated by	AssignFrameIdRange.framePid			
Attribute	Type	Mult.	Kind	Note
index	Integer	1	attr	This attribute is used to order the frame_PIDs. The values of index shall be unique within one AssignFrameIdRange.
pid	PositiveInteger	1	attr	Frame_PID value.

Table A.511: FramePid

Class	FramePort			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Connectors reception or send port on the referenced channel referenced by a FrameTriggering.			
Base	ARObject, CommConnectorPort, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	CommunicationConnector.ecuCommPortInstance			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.512: FramePort

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	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.
framePort	FramePort	*	ref	References to the FramePort on every ECU of the system which sends and/or receives the frame. References for both the sender and the receiver side shall be included when the system is completely defined.
pduTriggering	PduTriggering	*	ref	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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=pduTriggering.pduTriggering, pdu Triggering.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.513: FrameTriggering

Class	FunctionInhibitionNeeds			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	Specifies the abstract needs on the configuration of the Function Inhibition Manager for one Function Identifier (FID). This class currently contains no attributes. Its name can be regarded as a symbol identifying the FID from the viewpoint of the component or module which owns this class.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *ServiceNeeds*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.514: FunctionInhibitionNeeds

Class	GeneralPurposeConnection			
Package	M2::AUTOSARTemplates::SystemTemplate::GeneralPurposeConnection			
Note	This meta-class allows to describe the relationship between several PduTriggerings that are defined on the same PhysicalChannel, e.g. to create a link between Rx and Tx Pdu that are used for request/ response. **Tags:**atp.recommendedPackage=GeneralPurposeConnections			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
pduTriggering	PduTriggering	*	ref	Reference to PduTriggerings that are connected to each other by a GeneralPurposeConnection.

Table A.515: GeneralPurposeConnection

Class	GeneralPurposeIPdu			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	This element is used for AUTOSAR Pdus without attributes that are routed by the PduR. Please note that the category name of such Pdus is standardized in the AUTOSAR System Template. **Tags:**atp.recommendedPackage=Pdus			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *IPdu*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Pdu*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.516: GeneralPurposeIPdu

Class	GeneralPurposePdu			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	This element is used for AUTOSAR Pdus without additional attributes that are routed by a bus interface. Please note that the category name of such Pdus is standardized in the AUTOSAR System Template. **Tags:**atp.recommendedPackage=Pdus			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Pdu*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.517: GeneralPurposePdu

Class	GlobalTimeDomain			
Package	M2::AUTOSARTemplates::SystemTemplate::GlobalTime			
Note	This represents the ability to define a global time domain.			
	Tags:atp.recommendedPackage=GlobalTimeDomains			
Base	ARObject, CollectableElement, FibexElement, Identifiable, MultilanguageReferrable, Packageable Element, Referrable			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
debounceTime	TimeValue	0..1	attr	Defines the minimum amount of time between two time sync messages are transmitted.
domainId	PositiveInteger	1	attr	This represents the ID of the GlobalTimeDomain used in the network messages sent on behalf of global time management.
gateway	GlobalTimeGateway	*	aggr	A GlobalTimeGateway may exist in the context of a GlobalTimeDomain to actively update the global time information as it is routed from one GlobalTimeDomain to another.
				Stereotypes: atpSplitable; atpVariation **Tags:** atp.Splitkey=gateway.shortName, gateway.variation Point.shortLabel vh.latestBindingTime=postBuild
globalTime CorrectionProps	GlobalTimeCorrection Props	0..1	aggr	Defintion of attributes for rate and offset correction.
globalTime Domain Property	AbstractGlobalTime DomainProps	0..1	aggr	Additional properties of the GlobalTimeDomain.
				Stereotypes: atpSplitable; atpVariation **Tags:** atp.Splitkey=globalTimeDomainProperty, globalTime DomainProperty.variationPoint.shortLabel vh.latestBindingTime=postBuild
globalTime Master	GlobalTimeMaster	0..1	aggr	This represents the single master of a GlobalTime Domain. A GlobalTimeDomain may have no GlobalTime Domain.master, e.g. when it gets its time from a GPS receiver.
				Stereotypes: atpSplitable; atpVariation **Tags:** atp.Splitkey=globalTimeMaster.shortName, globalTime Master.variationPoint.shortLabel vh.latestBindingTime=postBuild
globalTimeSub Domain	GlobalTimeDomain	*	ref	By this means it is possible to create a hierarchy of sub Domains where one global time domain can declare one or more other global time domains as its subDomains.
				Stereotypes: atpSplitable; atpVariation **Tags:** atp.Splitkey=globalTimeSubDomain.globalTimeDomain, globalTimeSubDomain.variationPoint.shortLabel vh.latestBindingTime=postBuild
network SegmentId	NetworkSegment Identification	0..1	aggr	Defines the numerical identification of a GlobalTime sub domain.
offsetTime Domain	GlobalTimeDomain	0..1	ref	Reference to a synchronized time domain this offset time domain is based on. The reference source is the offset time domain. The reference target is the synchronized time domain.

▽

△

Class	GlobalTimeDomain			
pduTriggering	PduTriggering	0..1	ref	This PduTriggering will be taken to transmit the global time information from a GlobalTimeMaster to a the associated GlobalTimeSlaves. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=pduTriggering.pduTriggering, pduTriggering.variationPoint.shortLabel vh.latestBindingTime=postBuild
slave	GlobalTimeSlave	*	aggr	This represents the collections of slaves of the Global TimeDomain. A GlobalTimeDomain may have no Global TimeDomain.slaves, e.g. when it propagates its time directly to sub domains. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=slave.shortName, slave.variationPoint.short Label vh.latestBindingTime=postBuild
syncLoss Timeout	TimeValue	0..1	attr	This attribute describes the timeout for the situation that the time synchronization gets lost in the scope of the time domain.

Table A.518: GlobalTimeDomain

Class	GlobalTimeEthMaster			
Package	M2::AUTOSARTemplates::SystemTemplate::GlobalTime::ETH			
Note	This represents the specialization of the GlobalTimeMaster for Ethernet communication.			
Base	ARObject, GlobalTimeMaster, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	GlobalTimeDomain.globalTimeMaster			
Attribute	Type	Mult.	Kind	Note
crcSecured	GlobalTimeCrcSupport Enum	0..1	attr	Definition of whether or not CRC is supported. This is only relevant for selected bus systems.
subTlvConfig	EthTSynSubTlvConfig	0..1	aggr	Defines the subTLV fields which shall be included in the time sync message.

Table A.519: GlobalTimeEthMaster

Class	GlobalTimeGateway			
Package	M2::AUTOSARTemplates::SystemTemplate::GlobalTime			
Note	This represents the ability to define a time gateway for establishing a global time domain over several communication clusters.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	GlobalTimeDomain.gateway			
Attribute	Type	Mult.	Kind	Note
host	EcuInstance	1	ref	The GlobalTimeGateway is hosted by the referenced Ecu Instance.
master	GlobalTimeMaster	1	ref	This represents the master of the global time gateway.
slave	GlobalTimeSlave	1	ref	This represents the slave of the GlobalTimeGateway.

Table A.520: GlobalTimeGateway

Class	GlobalTimeMaster (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::GlobalTime			
Note	This represents the generic concept of a global time master.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Subclasses	GlobalTimeCanMaster, GlobalTimeEthMaster, GlobalTimeFrMaster, UserDefinedGlobalTimeMaster			
Aggregated by	GlobalTimeDomain.globalTimeMaster			
Attribute	Type	Mult.	Kind	Note
communication Connector	Communication Connector	1	ref	The GlobalTimeMaster is bound to the Communication Connector.
icvSecured	GlobalTimeIcvSupport Enum	0..1	attr	Defines whether an Integrity Check Value (ICV) shall be added to the sent time sync messages. **Tags:**atp.Status=candidate
immediate ResumeTime	TimeValue	0..1	attr	Defines the minimum time between an "immediate" message and the next periodic message.
isSystemWide GlobalTime Master	Boolean	1	attr	If set to TRUE, the GlobalTimeMaster is supposed to act as the root of global time information.
syncPeriod	TimeValue	1	attr	This represents the period. Unit: seconds

Table A.521: GlobalTimeMaster

Class	GlobalTimeSlave (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::GlobalTime			
Note	This represents the generic concept of a global time slave.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Subclasses	GlobalTimeCanSlave, GlobalTimeEthSlave, GlobalTimeFrSlave, UserDefinedGlobalTimeSlave			
Aggregated by	GlobalTimeDomain.slave			
Attribute	Type	Mult.	Kind	Note
communication Connector	Communication Connector	1	ref	The GlobalTimeSlave is bound to the Communication Connector.
followUp TimeoutValue	TimeValue	0..1	attr	Rx timeout for the follow-up message.
icvVerification	GlobalTimeIcv VerificationEnum	0..1	attr	Defines how an Integrity Check Value (ICV) shall be handled at the receiver. **Tags:**atp.Status=candidate
timeLeapFuture Threshold	TimeValue	0..1	attr	Defines the maximum allowed positive difference between the current Local Time Base value and a newly received Global Time Base value.
timeLeap HealingCounter	PositiveInteger	0..1	attr	Defines the required number of updates to the Time Base where the time difference to the previous received value has to remain within the bounds of timeLeapFuture Threshold and timeLeapPastThreshold until that Time Base is considered healed.
timeLeapPast Threshold	TimeValue	0..1	attr	Defines the maximum allowed negative difference between the current Local Time Base value and a newly received Global Time Base value.

Table A.522: GlobalTimeSlave

Enumeration	HandleInvalidEnum
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication
Note	Strategies of handling the reception of invalidValue.
Aggregated by	InvalidationPolicy.handleInvalid, ISignalPort.handleInvalid
Literal	Description
dontInvalidate	Invalidation is switched off. **Tags:**atp.EnumerationLiteralIndex=0
external Replacement	Replace a received invalidValue. The replacement value is sourced from the aggregation in the role replaceWith. **Tags:**atp.EnumerationLiteralIndex=1
keep	The application software is supposed to handle signal invalidation on RTE API level either by Data ReceiveErrorEvent or check of error code on read access. **Tags:**atp.EnumerationLiteralIndex=2
replace	Replace a received invalidValue. The replacement value is specified by the initValue. **Tags:**atp.EnumerationLiteralIndex=3

Table A.523: HandleInvalidEnum

Enumeration	HandleOutOfRangeEnum
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication
Note	A value of this type is taken for controlling the range checking behavior of the AUTOSAR RTE.
Aggregated by	ISignalProps.handleOutOfRange, ReceiverComSpec.handleOutOfRange, SenderComSpec.handle OutOfRange
Literal	Description
default	The RTE will use the initValue if the actual value is out of the specified bounds. **Tags:**atp.EnumerationLiteralIndex=0
external Replacement	This indicates that the value replacement is sourced from the attribute replaceWith. **Tags:**atp.EnumerationLiteralIndex=1
ignore	The RTE will ignore any attempt to send or receive the corresponding dataElement if the value is out of the specified range. **Tags:**atp.EnumerationLiteralIndex=2
invalid	The RTE will use the invalidValue if the value is out of the specified bounds. **Tags:**atp.EnumerationLiteralIndex=3
none	A range check is not required. **Tags:**atp.EnumerationLiteralIndex=4
saturate	The RTE will saturate the value of the dataElement such that it is limited to the applicable upper bound if it is greater than the upper bound. Consequently, it is limited to the applicable lower bound if the value is less than the lower bound. **Tags:**atp.EnumerationLiteralIndex=5

Table A.524: HandleOutOfRangeEnum

Class	HardwareConfiguration
Package	M2::AUTOSARTemplates::CommonStructure::ResourceConsumption
Note	Describes in which mode the hardware is operating while needing this resource consumption.
Base	ARObject

▽

△

Class	HardwareConfiguration			
Aggregated by	*ExecutionTime*.hardwareConfiguration, *HeapUsage*.hardwareConfiguration, *StackUsage*.hardwareConfiguration			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
additional Information	String	0..1	attr	Specifies additional information on the Hardware Configuration.
processorMode	String	0..1	attr	Specifies in which mode the processor is operating.
processor Speed	String	0..1	attr	Specifies the speed the processor is operating.

Table A.525: HardwareConfiguration

Class	HwAttributeDef			
Package	M2::AUTOSARTemplates::EcuResourceTemplate::HwElementCategory			
Note	This metaclass represents the ability to define a particular hardware attribute. The category of this element defines the type of the attributeValue. If the category is Enumeration the hwAttributeEnumerationLiterals specify the available literals.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	HwCategory.hwAttributeDef			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
hwAttribute Literal	HwAttributeLiteralDef	*	aggr	The available EnumerationLiterals of the Enumeration definition. Only applicable if the category of the HwAttributeDef equals Enumeration.
isRequired	Boolean	0..1	attr	This attribute specifies if the defined attribute value is required to be provided.
unit	Unit	0..1	ref	This association specifies the physical unit of the defined hardware attribute. This is optional due to the fact that there are textual attributes.

Table A.526: HwAttributeDef

Class	HwAttributeValue			
Package	M2::AUTOSARTemplates::EcuResourceTemplate::HwElementCategory			
Note	This metaclass represents the ability to assign a hardware attribute value. Note that v and vt are mutually exclusive.			
Base	*ARObject*			
Aggregated by	*HwDescriptionEntity*.hwAttributeValue			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
annotation	Annotation	0..1	aggr	Optional annotation that can be added to each HwAttributeValue.
hwAttributeDef	HwAttributeDef	0..1	ref	This association represents the definition of the particular hardware attribute value.
v	Numerical	0..1	attr	This represents a numerical hardware attribute value. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=systemDesignTime
vt	VerbatimString	0..1	attr	This represents a textual hardware attribute value.

Table A.527: HwAttributeValue

Class	**HwDescriptionEntity** (abstract)				
Package	M2::AUTOSARTemplates::EcuResourceTemplate				
Note	This meta-class represents the ability to describe a hardware entity.				
Base	*ARObject*, *Referrable*				
Subclasses	HwElement, HwPin, HwPinGroup, HwType				
Attribute	**Type**	**Mult.**	**Kind**	**Note**	
hwAttribute Value	HwAttributeValue	*	aggr	This aggregation represents a particular hardware attribute value. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=hwAttributeValue, hwAttributeValue.variation Point.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=50	
hwCategory	HwCategory	*	ref	One of the associations representing one particular category of the hardware entity. **Tags:**xml.sequenceOffset=30	
hwType	HwType	0..1	ref	This association is used to assign an optional HwType which contains the common attribute values for all occurences of this HwDescriptionEntity. Note that Hw Types can not be redefined and therefore shall not have a hwType reference.	

Table A.528: HwDescriptionEntity

Class	**HwElement**				
Package	M2::AUTOSARTemplates::EcuResourceTemplate				
Note	This represents the ability to describe Hardware Elements on an instance level. The particular types of hardware are distinguished by the category. This category determines the applicable attributes. The possible categories and attributes are defined in HwCategory. **Tags:**atp.recommendedPackage=HwElements				
Base	*ARElement*, *ARObject*, *CollectableElement*, *HwDescriptionEntity*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*				
Aggregated by	ARPackage.element				
Attribute	**Type**	**Mult.**	**Kind**	**Note**	
hwElement Connection	HwElementConnector	*	aggr	This represents one particular connection between two hardware elements. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=hwElementConnection, hwElement Connection.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=110	
hwPinGroup	HwPinGroup	*	aggr	This aggregation is used to describe the connection facilities of a hardware element. Note that hardware element has no pins but only pingroups. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=hwPinGroup.shortName, hwPin Group.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=90	

▽

△

Class	HwElement			
nestedElement	HwElement	*	ref	This association is used to establish hierarchies of hw elements. Note that one particular HwElement can be target of this association only once. I.e. multiple instantiation of the same HwElement is not supported (at any hierarchy level). **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=nestedElement.hwElement, nested Element.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=70

Table A.529: HwElement

Class	HwElementConnector			
Package	M2::AUTOSARTemplates::EcuResourceTemplate			
Note	This meta-class represents the ability to connect two hardware elements. The details of the connection can be refined by hwPinGroupConnection.			
Base	ARObject, Describable			
Aggregated by	HwElement.hwElementConnection			
Attribute	Type	Mult.	Kind	Note
hwElement	HwElement	*	ref	This association connects two hardware elements.
hwPin Connection	HwPinConnector	*	aggr	This represents one particular connection between two hardware pins. This connection shall be used if pin-to-pin-connection is to be described but no description of the connection between the hierarchical composition of HwPinGroups (using HwPinGroupConnector) is required. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=hwPinConnection, hwPin Connection.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=60
hwPinGroup Connection	HwPinGroupConnector	*	aggr	This represents one particular connection between two hardware pin groups. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=hwPinGroupConnection, hwPinGroup Connection.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=50

Table A.530: HwElementConnector

Class	HwPinConnector			
Package	M2::AUTOSARTemplates::EcuResourceTemplate			
Note	This meta-class represents the ability to connect two pins.			
Base	ARObject, Describable			
Aggregated by	HwElementConnector.hwPinConnection, HwPinGroupConnector.hwPinConnection			
Attribute	Type	Mult.	Kind	Note
hwPin	HwPin	*	ref	This association connects two hardware pins.

Table A.531: HwPinConnector

Class	HwPinGroupConnector				
Package	M2::AUTOSARTemplates::EcuResourceTemplate				
Note	This meta-class represents the ability to connect two pin groups.				
Base	*ARObject*, *Describable*				
Aggregated by	HwElementConnector.hwPinGroupConnection				
Attribute	**Type**	**Mult.**	**Kind**	**Note**	
hwPin Connection	HwPinConnector	*	aggr	This represents one particular connection between two hardware pins. The connected pins shall match the connection provided by the parent hwPinGroup Connection. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=hwPinConnection, hwPin Connection.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime	
hwPinGroup	HwPinGroup	*	ref	This association connects two hardware pin groups.	

Table A.532: HwPinGroupConnector

Class	HwPortMapping				
Package	M2::AUTOSARTemplates::SystemTemplate::ECUResourceMapping				
Note	HWPortMapping specifies the hwCommunicationPort (defined in the ECU Resource Template) to realize the specified CommunicationConnector in a physical topology.				
Base	*ARObject*				
Aggregated by	ECUMapping.hwPortMapping				
Attribute	**Type**	**Mult.**	**Kind**	**Note**	
communication Connector	Communication Connector	1	ref	Reference to the CommunicationConnector in the System Template	
hw Communication Port	HwPinGroup	1	ref	Reference to the HwPinPortGroup of category CommunicationPort. The connection to the Hw CommunicationController is described in the Ecu Resource Description.	

Table A.533: HwPortMapping

Class	HwType				
Package	M2::AUTOSARTemplates::EcuResourceTemplate::HwElementCategory				
Note	This represents the ability to describe Hardware types on an abstract level. The particular types of hardware are distinguished by the category. This category determines the applicable attributes. The possible categories and attributes are defined in HwCategory. **Tags:** atp.recommendedPackage=HwTypes				
Base	*ARElement*, *ARObject*, *CollectableElement*, *HwDescriptionEntity*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*				
Aggregated by	ARPackage.element				
Attribute	**Type**	**Mult.**	**Kind**	**Note**	
–	–	–	–	–	

Table A.534: HwType

Class	IPSecRule			
Package	M2::AUTOSARTemplates::SystemTemplate::SecureCommunication			
Note	This element defines an IPsec rule that describes communication traffic that is monitored, protected and filtered.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	IPSecConfig.ipSecRule			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
direction	Communication DirectionType	0..1	attr	This attribute defines the direction in which the traffic is monitored. If this attribute is not set a bidirectional traffic monitoring is assumed.
headerType	IPsecHeaderTypeEnum	0..1	attr	Header type specifying the IPsec security mechanism.
ipProtocol	IPsecIpProtocolEnum	0..1	attr	This attribute defines the relevant IP protocol used in the Security Policy Database (SPD) entry.
localCertificate	CryptoService Certificate	*	ref	This reference identifies the applicable certificate used for a local authentication.
localId	String	0..1	attr	This attribute defines how the local participant should be identified for authentication.
localPortRange End	PositiveInteger	0..1	attr	This attribute restricts the traffic monitoring and defines an end value for the local port range. If this attribute is not set then this rule shall be effective for all local ports. Please note that port ranges are currently not supported in the AUTOSAR AP's operating system backend. If AP systems are involved, each IPsec rule may only contain a single port.
localPortRange Start	PositiveInteger	0..1	attr	This attribute restricts the traffic monitoring and defines a start value for the local port range. If this attribute is not set then this rule shall be effective for all local ports. Please note that port ranges are currently not supported in the AUTOSAR AP's operating system backend. If AP systems are involved, each IPsec rule may only contain a single port.
mode	IPsecModeEnum	0..1	attr	This attribute defines the type of the connection.
policy	IPsecPolicyEnum	0..1	attr	An IPsec policy defines the rules that determine which type of IP traffic needs to be secured using IPsec and how that traffic is secured.
preSharedKey	CryptoServiceKey	0..1	ref	This reference identifies the applicable cryptograhic key used for authentication.
priority	PositiveInteger	0..1	attr	This attribute defines the priority of the IPSecRule (SPD entry). The processing of entries is based on priority, starting with the highest priority "0".
remote Certificate	CryptoService Certificate	*	ref	This reference identifies the applicable certificate used for a remote authentication.
remoteId	String	0..1	attr	This attribute defines how the remote participant should be identified for authentication.
remoteIp Address	NetworkEndpoint	*	ref	Definition of the remote NetworkEndpoint. With this reference the connection between the local Network Endpoint and the remote NetworkEndpoint is described on which the traffic is monitored.

$\bigtriangledown$

△

Class	IPSecRule			
remotePort RangeEnd	PositiveInteger	0..1	attr	This attribute restricts the traffic monitoring and defines an end value for the remote port range.
				If this attribute is not set then this rule shall be effective for all local ports.
				Please note that port ranges are currently not supported in the AUTOSAR AP's operating system backend. If AP systems are involved, each IPsec rule may only contain a single port.
remotePort RangeStart	PositiveInteger	0..1	attr	This attribute restricts the traffic monitoring and defines a start value for the remote port range.
				If this attribute is not set then this rule shall be effective for all local ports.
				Please note that port ranges are currently not supported in the AUTOSAR AP's operating system backend. If AP systems are involved, each IPsec rule may only contain a single port.

Table A.535: IPSecRule

Class	IPdu (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	ARObject, CollectableElement, FibexElement, Identifiable, MultilanguageReferrable, Packageable Element, Pdu, Referrable			
Subclasses	ContainerIPdu, DcmIPdu, GeneralPurposeIPdu, ISignalIPdu, J1939DcmIPdu, MultiplexedIPdu, NPdu, SecuredIPdu, UserDefinedIPdu			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
containedIPdu Props	ContainedIPduProps	0..1	aggr	Defines whether this IPdu may be collected inside a ContainerIPdu.

Table A.536: IPdu

Class	IPduMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Multiplatform			
Note	Arranges those IPdus that are transferred by the gateway from one channel to the other in pairs and defines the mapping between them.			
Base	ARObject			
Aggregated by	Gateway.iPduMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
introduction	DocumentationBlock	0..1	aggr	This represents introductory documentation about the IPdu mapping.
pduMaxLength	PositiveInteger	0..1	attr	Define the maximum length in bytes which limits the length of the Pdu during gateway operation if the runtime length of the received Pdu exceeds this limit.
pdurTpChunk Size	PositiveInteger	0..1	attr	Optionally defines the to be configured Pdu Router Tp ChunkSize for this routing relation.
sourceIPdu	PduTriggering	1	ref	Source destination of the referencing mapping.
targetIPdu	TargetIPduRef	1	aggr	Target destination of the referencing mapping.

Table A.537: IPduMapping

Class	IPduPort			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Connectors reception or send port on the referenced channel referenced by a PduTriggering.			
Base	*ARObject*, *CommConnectorPort*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*CommunicationConnector*.ecuCommPortInstance			
Attribute	Type	Mult.	Kind	Note
iPduSignal Processing	IPduSignalProcessing Enum	0..1	attr	Definition of the two signal processing modes Immediate and Deferred for both Tx and Rx IPdus.
rxSecurity Verification	Boolean	0..1	attr	This attribute defines the bypassing of signature authentication or MAC verification in the receiving ECU. If not defined or set to true the signature authentication or MAC verification shall be performed for the SecuredIPdu. If set to false the signature authentication or MAC verification shall not be performed for the SecuredIPdu.
timestampRx Acceptance Window	TimeValue	0..1	attr	This attribute is used to define the maximum allowed deviation in seconds from the expected timestamp for which a SecuredIPdu is still deemed authentic. Please note that this attribute is for documentation only to allow the configuration of required freshness value manager and no upstream mapping is defined for it.
useAuthData Freshness	Boolean	0..1	attr	This attribute describes whether a part of AuthenticPdu contained in a SecuredIPdu shall be passed on to the SWC that verifies and generates the Freshness. The part of the Authentic-PDU is defined by the authData FreshnessStartPosition and authDataFreshnessLength.

Table A.538: IPduPort

Class	IPv6ExtHeaderFilterList			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::IPv6HeaderFilterList			
Note	Permitted list for the filtering of IPv6 extension headers.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	IPv6ExtHeaderFilterSet.extHeaderFilterList			
Attribute	Type	Mult.	Kind	Note
allowedIPv6Ext Header	PositiveInteger	1..*	attr	IPv6 Extension Header type allowed by this filter.

Table A.539: IPv6ExtHeaderFilterList

Class	ISignal
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication
Note	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. 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. ISignals describe the Interface between the Precompile configured RTE and the potentially Postbuild configured Com Stack (see ECUC Parameter Mapping). In case of the SystemSignalGroup an ISignal shall be created for each SystemSignal contained in the SystemSignalGroup. **Tags:**atp.recommendedPackage=ISignals
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*
Aggregated by	ARPackage.element

▽

△

Class	**ISignal**			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
data Transformation	DataTransformation	0..1	ref	Optional reference to a DataTransformation which represents the transformer chain that is used to transform the data that shall be placed inside this ISignal. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=dataTransformation.dataTransformation, dataTransformation.variationPoint.shortLabel vh.latestBindingTime=codeGenerationTime
dataTypePolicy	DataTypePolicyEnum	1	attr	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. 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.
initValue	ValueSpecification	0..1	aggr	Optional definition of a ISignal's initValue in case the System Description doesn't use a complete Software Component Description (VFB View). This supports the inclusion of legacy system signals. This value can be used to configure the Signal's "Init Value". 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.
iSignalProps	ISignalProps	0..1	aggr	Additional optional ISignal properties that may be stored in different files. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=iSignalProps
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	1	attr	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. The ISignal length of zero bits is allowed.

▽

△

Class	ISignal			
network Representation Props	SwDataDefProps	0..1	aggr	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 "memAllignment" and "byteOrder" shall not be used.
				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.
				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.
				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.
				Stereotypes: atpSplitable **Tags:**atp.Splitkey=networkRepresentationProps
systemSignal	SystemSignal	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 ComTimeoutSubstituition for this ISignal.
transformation ISignalProps	TransformationISignal Props	*	aggr	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.
				Stereotypes: atpSplitable **Tags:**atp.Splitkey=transformationISignalProps

Table A.540: ISignal

Class	ISignalGroup			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	SignalGroup of the Interaction Layer. The RTE supports a "signal fan-out" where the same System Signal Group is sent in different SignalIPdus to multiple receivers.			
	An ISignalGroup refers to a set of ISignals that shall always be kept together. A ISignalGroup represents a COM Signal Group.			
	Therefore it is recommended to put the ISignalGroup in the same Package as ISignals (see atp.recommendedPackage)			
	Tags:atp.recommendedPackage=ISignalGroup			
Base	ARObject, CollectableElement, FibexElement, Identifiable, MultilanguageReferrable, Packageable Element, Referrable			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note

▽

△

Class	**ISignalGroup**			
comBased SignalGroup Transformation	DataTransformation	0..1	ref	Optional reference to a DataTransformation which represents the transformer chain that is used to transform the data that shall be placed inside this ISignalGroup based on the COMBasedTransformer approach. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=comBasedSignalGroupTransformation.data Transformation, comBasedSignalGroup Transformation.variationPoint.shortLabel vh.latestBindingTime=codeGenerationTime
iSignal	ISignal	*	ref	Reference to a set of ISignals that shall always be kept together.
systemSignal Group	SystemSignalGroup	1	ref	Reference to the SystemSignalGroup that is defined on VFB level and that is supposed to be transmitted in the ISignalGroup.
transformation ISignalProps	TransformationISignal Props	*	aggr	A transformer chain consists of an ordered list of transformers. The ISignalGroup specific configuration properties for each transformer are defined in the TransformationISignalProps class. The transformer configuration properties that are common for all ISignal Groups are described in the TransformationTechnology class. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=transformationISignalProps

Table A.541: ISignalGroup

Class	**ISignalIPdu**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Represents the IPdus handled by Com. The ISignalIPdu assembled and disassembled in AUTOSAR COM consists of one or more signals. In case no multiplexing is performed this IPdu is routed to/from the Interface Layer. A maximum of one dynamic length signal per IPdu is allowed. **Tags:**atp.recommendedPackage=Pdus			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *IPdu*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Pdu*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
iPduTiming Specification	IPduTiming	0..1	aggr	Timing specification for Com IPdus (Transmission Modes). This information is mandatory for the sender in a System Extract. This information may be omitted on receivers in a System Extract. atpVariation: The timing of a Pdu can vary. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=iPduTimingSpecification, iPduTiming Specification.variationPoint.shortLabel vh.latestBindingTime=postBuild

▽

△

Class	ISignalIPdu				
iSignalToPdu Mapping	ISignalToIPduMapping	*	aggr	Definition of SignalToIPduMappings included in the Signal IPdu. atpVariation: The content of a PDU can be variable. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=iSignalToPduMapping.shortName, iSignalTo PduMapping.variationPoint.shortLabel vh.latestBindingTime=postBuild	
unusedBit Pattern	Integer	1	attr	AUTOSAR COM and AUTOSAR IPDUM are filling not used areas of an IPDU with this bit-pattern. This attribute is mandatory to avoid undefined behavior. This byte-pattern will be repeated throughout the IPdu.	

Table A.542: ISignalIPdu

Class	ISignalIPduGroup			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	The AUTOSAR COM Layer is able to start and to stop sending and receiving configurable groups of I-Pdus during runtime. An ISignalIPduGroup contains either ISignalIPdus or ISignalIPduGroups. **Tags:**atp.recommendedPackage=ISignaliPduGroup			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
communication Direction	Communication DirectionType	1	attr	This attribute determines in which direction IPdus that are contained in this IPduGroup will be transmitted (communication direction can be either In or Out).
communication Mode	String	1	attr	This attribute defines the use-case for this ISignalIPdu Group (e.g. diagnostic, debugging etc.). For example, in a diagnostic mode all IPdus - which are not involved in diagnostic - are disabled. The use cases are not limited to a fixed enumeration and can be specified as a string.
contained ISignalIPdu Group	ISignalIPduGroup	*	ref	An I-Pdu group can be included in other I-Pdu groups. Contained I-Pdu groups shall not be referenced by the EcuInstance.
iSignalIPdu	ISignalIPdu	*	ref	Reference to a set of Signal I-Pdus, which are contained in the ISignal I-Pdu Group. atpVariation: The content of a ISignal I-Pdu group can vary (->vehicle modes). **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=iSignalIPdu.iSignalIPdu, iSignal IPdu.variationPoint.shortLabel vh.latestBindingTime=postBuild
nmPdu	NmPdu	*	ref	Reference to a set of NmPdus with NmUserData, which are contained in the ISignalIPduGroup. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=nmPdu.nmPdu, nmPdu.variationPoint.short Label vh.latestBindingTime=postBuild

Table A.543: ISignalIPduGroup

Class	ISignalMapping
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Multiplatform
Note	Arranges those signals (or SignalGroups) that are transferred by the gateway from one channel to the other in pairs and defines the mapping between them. Each pair consists in a source and a target referencing to a ISignalTriggering.
Base	ARObject
Aggregated by	Gateway.signalMapping

Attribute	Type	Mult.	Kind	Note
introduction	DocumentationBlock	0..1	aggr	This represents introductory documentation about the ISignal mapping.
sourceSignal	ISignalTriggering	1	ref	Source destination of the referencing mapping.
targetSignal	ISignalTriggering	1	ref	Target destination of the referencing mapping.

Table A.544: ISignalMapping

Class	ISignalPort
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication
Note	Connectors reception or send port on the referenced channel referenced by an ISignalTriggering. If different timeouts or DataFilters for ISignals need to be specified several ISignalPorts may be created.
Base	ARObject, CommConnectorPort, Identifiable, MultilanguageReferrable, Referrable
Aggregated by	CommunicationConnector.ecuCommPortInstance

Attribute	Type	Mult.	Kind	Note
dataFilter	DataFilter	0..1	aggr	Optional specification of a signal COM filter at the receiver side in case that the System Description doesn't use a complete Software Component Description (VFB View). This supports the inclusion of legacy system signals. If a full DataMapping exist for the SystemSignal this information may be available from a configured ReceiverComSpec. In this case the ReceiverComSpec overrides this optional specification.
firstTimeout	TimeValue	0..1	attr	• ISignalPort with communicationDirection = in: Optional first timeout value in seconds for the reception of the ISignal. • ISignalPort with communicationDirection = out: Optional first timeout value in seconds for transmission deadline monitoring.
handleInvalid	HandleInvalidEnum	0..1	attr	This attribute defines how invalidation is applied to the ISignals received in the context of this ISignalPort.
timeout	TimeValue	0..1	attr	• ISignalPort with communicationDirection = in: Optional timeout value in seconds for the reception of the ISignal. The attribute value is used to configure the Com Timeout in the COM module. The RTE ignores this attribute. The timeout can also be specified with the NonqueuedReceiverComSpec.aliveTimeout attribute. If a full DataMapping exists for the SystemSignal and the value is available in the configured ReceiverComSpec, then the timeout value in the ReceiverComSpec overrides this optional timeout specification during the creation of the Base Ecu Configuration of the COM module. • ISignalPort with communicationDirection = out: Optional timeout value in seconds for the transmission of the ISignal. The attribute value is used to configure the ComTimeout in the COM module. The RTE ignores this

▽

Document ID 635: AUTOSAR_TR_ClassicAutosarModelConstraints

△

Class	ISignalPort			
				△ attribute. The timeout can also be specified with the ender ComSpec.transmissionAcknowledge.timeout attribute. If a full DataMapping exists for the SystemSignal and the value is available in the configured SenderComSpec, then the timeout value in the SenderComSpec overrides this optional timeout specification during the creation of the Base Ecu Configuration of the COM module. This attribute can be used in the following cases: - legacy signal where the System Description doesn't use a complete Software Component Description (VFB View) and where the Data Mapping is missing. - bus monitoring use cases in which the Data Mapping is ignored.

Table A.545: ISignalPort

Class	ISignalToIPduMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	An ISignalToIPduMapping describes the mapping of ISignals to ISignalIPdus and defines the position of the ISignal within an ISignalIPdu.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ISignalIPdu.iSignalToPduMapping, NmPdu.iSignalToIPduMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
iSignal	ISignal	0..1	ref	Reference to a ISignal that is mapped into the ISignal IPdu. Each ISignal contained in the ISignalGroup shall be mapped into an IPdu by an own ISignalToIPduMapping. The references to the ISignal and to the ISignalGroup in an ISignalToIPduMapping are mutually exclusive.
iSignalGroup	ISignalGroup	0..1	ref	Reference to an ISignalGroup that is mapped into the SignalIPdu. If an ISignalToIPduMapping for an ISignal Group is defined, only the UpdateIndicationBitPosition and the transferProperty is relevant. The startPosition and the packingByteOrder shall be ignored. Each ISignal contained in the ISignalGroup shall be mapped into an IPdu by an own ISignalToIPduMapping. The references to the ISignal and to the ISignalGroup in an ISignalToIPduMapping are mutually exclusive.
packingByte Order	ByteOrderEnum	0..1	attr	This parameter defines the order of the bytes of the signal and the packing into the SignalIPdu. The byte ordering "Little Endian" (MostSignificantByteLast), "Big Endian" (MostSignificantByteFirst) and "Opaque" can be selected. For opaque data endianness conversion shall be configured to Opaque. The value of this attribute impacts the absolute position of the signal into the SignalIPdu (see the startPosition attribute description). For an ISignalGroup the packingByteOrder is irrelevant and shall be ignored.

▽

△

Class	ISignalToIPduMapping			
startPosition	UnlimitedInteger	0..1	attr	This parameter is necessary to describe the bitposition of a signal within an SignalIPdu. It denotes the least significant bit for "Little Endian" and the most significant bit for "Big Endian" packed signals within the IPdu (see the description of the packingByteOrder attribute). In AUTOSAR the bit counting is always set to "sawtooth" and the bit order is set to "Decreasing". The bit counting in byte 0 starts with bit 0 (least significant bit). The most significant bit in byte 0 is bit 7. Please note that the way the bytes will be actually sent on the bus does not impact this representation: they will always be seen by the software as a byte array. If a mapping for the ISignalGroup is defined, this attribute is irrelevant and shall be ignored.
transferProperty	TransferPropertyEnum	0..1	attr	Defines how the referenced ISignal contributes to the send triggering of the ISignalIPdu.
update IndicationBit Position	UnlimitedInteger	0..1	attr	The UpdateIndicationBit indicates to the receivers that the signal (or the signal group) was updated by the sender. Length is always one bit. The UpdateIndicationBitPosition attribute describes the position of the update bit within the SignalIPdu. For Signals of a ISignalGroup this attribute is irrelevant and shall be ignored. Note that the exact bit position of the updateIndicationBit Position is linked to the value of the attribute packingByte Order because the method of finding the bit position is different for the values mostSignificantByteFirst and most SignificantByteLast. This means that if the value of packingByteOrder is changed while the value of update IndicationBitPosition remains unchanged the exact bit position of updateIndicationBitPosition within the enclosing ISignalIPdu still undergoes a change. This attribute denotes the least significant bit for "Little Endian" and the most significant bit for "Big Endian" packed signals within the IPdu (see the description of the packingByteOrder attribute). In AUTOSAR the bit counting is always set to "sawtooth" and the bit order is set to "Decreasing". The bit counting in byte 0 starts with bit 0 (least significant bit). The most significant bit in byte 0 is bit 7.

Table A.546: ISignalToIPduMapping

Class	ISignalTriggering			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	A ISignalTriggering allows an assignment of ISignals to physical channels.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	PhysicalChannel.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 A.547: ISignalTriggering

Enumeration	ISignalTypeEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication
Note	This enumeration defines ISignal types that are used for derivation of the ComSignalType in the COM configuration.
Aggregated by	ISignal.iSignalType
Literal	Description
array	ISignal shall be interpreted as an array (UINT8_N, UINT8_DYN) **Tags:**atp.EnumerationLiteralIndex=0
primitive	ISignal shall be interpreted as a primitive type (e.g. UINT_8, SINT_32) **Tags:**atp.EnumerationLiteralIndex=1

Table A.548: ISignalTypeEnum

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, Referrable
Subclasses	ARPackage, AbstractDoIpLogicAddressProps, AbstractEvent, AbstractImplementationDataTypeElement, AbstractSecurityEventFilter, AbstractSecurityIdsmInstanceFilter, AbstractServiceInstance, AppOsTask ProxyToEcuTaskProxyMapping, ApplicationEndpoint, ApplicationError, ApplicationPartitionToEcuPartition Mapping, AsynchronousServerCallResultPoint, AtpBlueprint, AtpBlueprintable, AtpClassifier, Atp Feature, AutosarOperationArgumentInstance, AutosarVariableInstance, BinaryManifestAddressable Object, BinaryManifestItemDefinition, BinaryManifestResource, BinaryManifestResourceDefinition, BlockState, BswInternalTriggeringPoint, BswModuleDependency, BuildActionEntity, BuildAction Environment, CanTpAddress, CanTpChannel, CanTpNode, Chapter, ClassContentConditional, ClientId Definition, ClientServerOperation, Code, CollectableElement, ComManagementMapping, Comm ConnectorPort, CommunicationConnector, CommunicationController, Compiler, ConsistencyNeeds, ConsumedEventGroup, CouplingPort, CouplingPortStructuralElement, CpSoftwareClusterResource, Cp SoftwareClusterResourceToApplicationPartitionMapping, CpSoftwareClusterToEcuInstanceMapping, Cp SoftwareClusterToResourceMapping, CryptoServiceMapping, DataPrototypeGroup, DataTransformation, DependencyOnArtifact, DiagEventDebounceAlgorithm, DiagnosticConnectedIndicator, DiagnosticData Element, DiagnosticDebounceAlgorithmProps, DiagnosticFunctionInhibitSource, DiagnosticParameter Element, DiagnosticRoutineSubfunction, DltApplication, DltArgument, DltLogChannel, DltMessage, DoIp Interface, DoIpLogicAddress, DoIpRoutingActivation, ECUMapping, EOCExecutableEntityRefAbstract, EcuPartition, EcucContainerValue, EcucDefinitionElement, EcucDestinationUriDef, EcucEnumeration LiteralDef, EcucQuery, EcucValidationCondition, EndToEndProtection, EthernetWakeupSleepOnDataline Config, EventHandler, ExclusiveArea, ExecutableEntity, ExecutionTime, FMAttributeDef, FMFeatureMap Assertion, FMFeatureMapCondition, FMFeatureMapElement, FMFeatureRelation, FMFeatureRestriction, FMFeatureSelection, FlatInstanceDescriptor, FlexrayArTpNode, FlexrayTpConnectionControl, FlexrayTp Node, FlexrayTpPduPool, FrameTriggering, GeneralParameter, GlobalTimeGateway, GlobalTimeMaster, GlobalTimeSlave, HeapUsage, HwAttributeDef, HwAttributeLiteralDef, HwPin, HwPinGroup, IPSecRule, IPv6ExtHeaderFilterList, ISignalToIPduMapping, ISignalTriggering, IdentCaption, InternalTriggeringPoint, J1939SharedAddressCluster, J1939TpNode, Keyword, LifeCycleState, LinScheduleTable, LinTpNode, Linker, MacMulticastGroup, MacSecKayParticipant, McDataInstance, MemorySection, ModeDeclaration,

▽

△

Class	Identifiable (abstract)			
	△ ModeDeclarationMapping, ModeSwitchPoint, NetworkEndpoint, *NmCluster*, NmEcu, *NmNode*, NvBlock Descriptor, *PackageableElement*, ParameterAccess, PduActivationRoutingGroup, PduToFrameMapping, PduTriggering, PerInstanceMemory, *PhysicalChannel*, PortElementToCommunicationResourceMapping, PortGroup, *PortInterfaceMapping*, PossibleErrorReaction, ResourceConsumption, RootSwComposition Prototype, RptComponent, RptContainer, RptExecutableEntity, RptExecutableEntityEvent, RptExecution Context, RptProfile, RptServicePoint, RteEventInCompositionSeparation, RteEventInCompositionToOs TaskProxyMapping, RteEventInSystemSeparation, RteEventInSystemToOsTaskProxyMapping, RunnableEntityGroup, *SdgAttribute*, SdgClass, SecureCommunicationAuthenticationProps, Secure CommunicationFreshnessProps, SecurityEventContextProps, *ServerCallPoint*, *ServiceNeeds*, Signal ServiceTranslationElementProps, SignalServiceTranslationEventProps, SignalServiceTranslationProps, SocketAddress, SomeipTpChannel, *SpecElementReference*, *StackUsage*, StaticSocketConnection, StructuredReq, SwGenericAxisParamType, SwServiceArg, SwcServiceDependency, SwcToApplication PartitionMapping, SwcToEcuMapping, SwcToImplMapping, SystemMapping, TDCpSoftwareCluster Mapping, TDCpSoftwareClusterResourceMapping, TcpOptionFilterList, *TimingClock*, TimingClockSync Accuracy, TimingCondition, *TimingConstraint*, *TimingDescription*, TimingExtensionResource, Timing ModeInstance, TlsCryptoCipherSuite, TlsCryptoCipherSuiteProps, Topic1, TpAddress, TraceableTable, TraceableText, *TracedFailure*, *TransformationProps*, TransformationTechnology, Trigger, VariableAccess, VariationPointProxy, ViewMap, VlanConfig, WaitPoint			
Attribute	Type	Mult.	Kind	Note
adminData	AdminData	0..1	aggr	This represents the administrative data for the identifiable object. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=adminData xml.sequenceOffset=-40
annotation	Annotation	*	aggr	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. **Tags:**xml.sequenceOffset=-25
category	CategoryString	0..1	attr	The category is a keyword that specializes the semantics of the Identifiable. It affects the expected existence of attributes and the applicability of constraints. **Tags:**xml.sequenceOffset=-50
desc	MultiLanguageOverview Paragraph	0..1	aggr	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. More elaborate documentation, (in particular how the object is built or used) should go to "introduction". **Tags:**xml.sequenceOffset=-60
introduction	DocumentationBlock	0..1	aggr	This represents more information about how the object in question is built or is used. Therefore it is a DocumentationBlock. **Tags:**xml.sequenceOffset=-30

▽

△

Class	Identifiable (abstract)			
uuid	String	0..1	attr	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. **Tags:**xml.attribute=true

Table A.549: Identifiable

Primitive	Identifier			
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::PrimitiveTypes			
Note	An Identifier is a string with a number of constraints on its appearance, satisfying the requirements typical programming languages define for their Identifiers. This datatype represents a string, that can be used as a c-Identifier. It shall start with a letter, may consist of letters, digits and underscores. **Tags:** xml.xsd.customType=IDENTIFIER xml.xsd.maxLength=128 xml.xsd.pattern=[a-zA-Z][a-zA-Z0-9_]* xml.xsd.type=string			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
blueprintValue	String	0..1	attr	This represents a description that documents how the value shall be defined when deriving objects from the blueprint. **Tags:** atp.Status=draft xml.attribute=true
namePattern	String	0..1	attr	This attribute represents a pattern which shall be used to define the value of the identifier if the identifier in question is part of a blueprint. For more details refer to TPS_StandardizationTemplate. **Tags:**xml.attribute=true

Table A.550: Identifier

Class	Implementation (abstract)			
Package	M2::AUTOSARTemplates::CommonStructure::Implementation			
Note	Description of an implementation a single software component or module.			
Base	ARElement, ARObject, CollectableElement, Identifiable, MultilanguageReferrable, Packageable Element, Referrable			
Subclasses	BswImplementation, SwcImplementation			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
buildAction Manifest	BuildActionManifest	0..1	ref	A manifest specifying the intended build actions for the software delivered with this implementation. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=buildActionManifest.buildActionManifest, buildActionManifest.variationPoint.shortLabel vh.latestBindingTime=codeGenerationTime
codeDescriptor	Code	*	aggr	Specifies the provided implementation code.
compiler	Compiler	*	aggr	Specifies the compiler for which this implementation has been released
generated Artifact	DependencyOnArtifact	*	aggr	Relates to an artifact that will be generated during the integration of this Implementation by an associated generator tool. Note that this is an optional information since it might not always be in the scope of a single module or component to provide this information. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=generatedArtifact.shortName, generated Artifact.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
hwElement	HwElement	*	ref	The hardware elements (e.g. the processor) required for this implementation.
linker	Linker	*	aggr	Specifies the linker for which this implementation has been released.
mcSupport	McSupportData	0..1	aggr	The measurement & calibration support data belonging to this implementation. The aggregtion is <<atpSplitable>> because in case of an already exisiting BSW Implementation model, this description will be added later in the process, namely at code generation time. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=mcSupport
programming Language	Programminglanguage Enum	0..1	attr	Programming language the implementation was created in.
requiredArtifact	DependencyOnArtifact	*	aggr	Specifies that this Implementation depends on the existance of another artifact (e.g. a library). This aggregation of DependencyOnArtifact is subject to variability with the purpose to support variability in the implementations. Different algorithms in the implementation might cause different dependencies, e.g. the number of used libraries. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=requiredArtifact.shortName, required Artifact.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	Implementation (abstract)			
required GeneratorTool	DependencyOnArtifact	*	aggr	Relates this Implementation to a generator tool in order to generate additional artifacts during integration. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=requiredGeneratorTool.shortName, required GeneratorTool.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
resource Consumption	ResourceConsumption	0..1	aggr	All static and dynamic resources for each implementation are described within the ResourceConsumption class. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=resourceConsumption.shortName
swcBsw Mapping	SwcBswMapping	0..1	ref	This allows a mapping between an SWC and a BSW behavior to be attached to an implementation description (for AUTOSAR Service, ECU Abstraction and Complex Driver Components). It is up to the methodology to define whether this reference has to be set for the Swc- or Bsw Implementtion or for both.
swVersion	RevisionLabelString	0..1	attr	Software version of this implementation. The numbering contains three levels (like major, minor, patch), its values are vendor specific.
usedCode Generator	String	0..1	attr	Optional: code generator used.
vendorId	PositiveInteger	0..1	attr	Vendor ID of this Implementation according to the AUTOSAR vendor list

Table A.551: Implementation

Class	ImplementationDataType			
Package	M2::AUTOSARTemplates::CommonStructure::ImplementationDataTypes			
Note	Describes a reusable data type on the implementation level. This will typically correspond to a typedef in C-code. **Tags:**atp.recommendedPackage=ImplementationDataTypes			
Base	*ARElement*, *ARObject*, *AbstractImplementationDataType*, *AtpBlueprint*, *AtpBlueprintable*, *AtpClassifier*, *AtpType*, *AutosarDataType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dynamicArray SizeProfile	String	0..1	attr	Specifies the profile which the array will follow in case this data type is a variable size array.
isStructWith Optional Element	Boolean	0..1	attr	This attribute is only valid if the attribute category is set to STRUCTURE. If set to true, this attribute indicates that the ImplementationDataType has been created with the intention to define at least one element of the structure as optional.

▽

△

Class	ImplementationDataType			
subElement (ordered)	ImplementationData TypeElement	*	aggr	Specifies an element of an array, struct, or union data type. The aggregation of ImplementionDataTypeElement is subject to variability with the purpose to support the conditional existence of elements inside a Implementation DataType representing a structure. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=subElement.shortName, sub Element.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
symbolProps	SymbolProps	0..1	aggr	This represents the SymbolProps for the Implementation DataType. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=symbolProps.shortName
typeEmitter	NameToken	0..1	attr	This attribute is used to control which part of the AUTOSAR toolchain is supposed to trigger data type definitions.

Table A.552: ImplementationDataType

Class	ImplementationDataTypeElement			
Package	M2::AUTOSARTemplates::CommonStructure::ImplementationDataTypes			
Note	Declares a data object which is locally aggregated. Such an element can only be used within the scope where it is aggregated. This element either consists of further subElements or it is further defined via its swDataDefProps. There are several use cases within the system of ImplementationDataTypes fur such a local declaration: • It can represent the elements of an array, defining the element type and array size • It can represent an element of a struct, defining its type • It can be the local declaration of a debug element.			
Base	ARObject, AbstractImplementationDataTypeElement, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	AtpClassifier.atpFeature, ImplementationDataType.subElement, ImplementationDataTypeElement.sub Element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
arrayImplPolicy	ArrayImplPolicyEnum	0..1	attr	This attribute controls the implementation of the payload of an array. It shall only be used if the enclosing ImplementationDataType constitutes an array.
arraySize	PositiveInteger	0..1	attr	The existence of this attributes (if bigger than 0) defines the size of an array and declares that this Implementation DataTypeElement represents the type of each single array element. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime
arraySize Handling	ArraySizeHandling Enum	0..1	attr	The way how the size of the array is handled in case of a variable size array.
arraySize Semantics	ArraySizeSemantics Enum	0..1	attr	This attribute controls the meaning of the value of the array size.

▽

△

Class	ImplementationDataTypeElement			
isOptional	Boolean	0..1	attr	This attribute represents the ability to declare the enclosing ImplementationDataTypeElement as optional. This means that, at runtime, the ImplementationDataType Element may or may not have a valid value and shall therefore be ignored. The underlying runtime software provides means to set the CppImplementationDataTypeElement as not valid at the sending end of a communication and determine its validity at the receiving end.
subElement (ordered)	ImplementationData TypeElement	*	aggr	Element of an array, struct, or union in case of a nested declaration (i.e. without using "typedefs"). The aggregation of ImplementionDataTypeElement is subject to variability with the purpose to support the conditional existence of elements inside a Implementation DataType representing a structure. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=subElement.shortName, sub Element.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
swDataDef Props	SwDataDefProps	0..1	aggr	The properties of this ImplementationDataTypeElement. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=swDataDefProps

Table A.553: ImplementationDataTypeElement

Class	ImplementationDataTypeElementInPortInterfaceRef			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer::InstanceRef			
Note	This meta-class represents the ability to refer to the internal structure of an AutosarDataPrototype which is typed by an ImplementationDatatype in the context of a PortInterface. In other words, this meta-class shall not be used to model a reference to the AutosarDataPrototype as a target itself, even if the AutosarDataPrototype is typed by an ImplementationDataType and even if that ImplementationDataType represents a composite data type.			
Base	ARObject, DataPrototypeReference			
Aggregated by	DataPrototypeTransformationProps.dataPrototypeInPortInterfaceRef, SignalServiceTranslationElement Props.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
context Implementation DataElement (ordered)	AbstractImplementation DataTypeElement	*	ref	This is a context in case there are subelements with explicit types. The reference has to be ordered to properly reflect the nested structure. **Tags:**xml.sequenceOffset=20
rootData Prototype	AutosarDataPrototype	0..1	ref	This refers to the AutosarDataPrototype which is typed by the ImplementationDatatype. The targetDataPrototype and all defined contextDataPrototypes can be found within this rootDataPrototype. **Tags:**xml.sequenceOffset=10
target Implementation DataType Element	AbstractImplementation DataTypeElement	0..1	ref	This is a target ImplementationDataTypeElement in case that the rootDataPrototype is composite and the target is a subElement of the rootDataPrototype. **Tags:**xml.sequenceOffset=30

Table A.554: ImplementationDataTypeElementInPortInterfaceRef

Class	ImplementationDataTypeSubElementRef			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	This meta-class represents the specialization of SubElementMapping with respect to Implementation DataTypes.			
Base	ARObject, SubElementRef			
Aggregated by	SubElementMapping.firstElement, SubElementMapping.secondElement			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
implementation DataType Element	ArVariableIn ImplementationData InstanceRef	0..1	aggr	This represents the referenced implementationDataType Element.
parameter Implementation DataType Element	ArParameterIn ImplementationData InstanceRef	0..1	aggr	This represents the referenced ImplementationDataType Element.

Table A.555: ImplementationDataTypeSubElementRef

Class	ImplementationElementInParameterInstanceRef			
Package	M2::AUTOSARTemplates::CommonStructure::MeasurementCalibrationSupport			
Note	Describes a reference to a particular ImplementationDataTypeElement instance in the context of a given ParameterDataPrototype. Thus it refers to a particular element in the implementation description of a software data structure.			
	Use Case: The RTE generator publishes its generated structure of calibration parameters in its BSW module description using the "constantMemory" role of ParameterDataPrototypes. Each ParameterData Prototype describes a group of single calibration parameters. In order to point to these single parameters, this "instance ref" is needed.			
	Note that this class follows the pattern of an InstanceRef but is not implemented based on the abstract classes because the ImplementationDataType isn't either, especially because ImplementationDataType Element isn't derived from AtpPrototype.			
Base	ARObject			
Aggregated by	McDataInstance.instanceInMemory			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
context	ParameterData Prototype	0..1	ref	The context for the referred element. **Tags:**xml.sequenceOffset=20
target	AbstractImplementation DataTypeElement	0..1	ref	The referred data element. **Tags:**xml.sequenceOffset=30

Table A.556: ImplementationElementInParameterInstanceRef

Class	ImplementationProps (abstract)			
Package	M2::AUTOSARTemplates::CommonStructure::Implementation			
Note	Defines a symbol to be used as (depending on the concrete case) either a complete replacement or a prefix when generating code artifacts.			
Base	ARObject, Referrable			
Subclasses	BswSchedulerNamePrefix, ExecutableEntityActivationReason, SectionNamePrefix, SymbolProps, SymbolicNameProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
symbol	CIdentifier	0..1	attr	The symbol to be used as (depending on the concrete case) either a complete replacement or a prefix.

Table A.557: ImplementationProps

Class	IndexedArrayElement			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	This element represents exactly one indexed element in the array. Either the applicationArrayElement or implementationArrayElement reference shall be used.			
Base	ARObject			
Aggregated by	SenderRecArrayElementMapping.indexedArrayElement			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
applicationArray Element	ApplicationArray Element	0..1	ref	Reference to an ApplicationArrayElement in an array.
implementation ArrayElement	ImplementationData TypeElement	0..1	ref	Reference to an ImplementationDataTypeElement in an array.
index	Integer	1	attr	Position of an element in an array. Starting position is 0.

Table A.558: IndexedArrayElement

Class	InitEvent			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents			
Note	This RTEEvent is supposed to be used for initialization purposes, i.e. for starting and restarting a partition. It is not guaranteed that all RunnableEntities referenced by this InitEvent are executed before the 'regular' RunnableEntities are executed for the first time. The execution order depends on the task mapping.			
Base	ARObject, AbstractEvent, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable, Multilanguage Referrable, RTEEvent, Referrable			
Aggregated by	AtpClassifier.atpFeature, SwcInternalBehavior.event			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.559: InitEvent

Class	InstantiationDataDefProps			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::InstantiationDataDefProps			
Note	This is a general class allowing to apply additional SwDataDefProps to particular instantiations of a Data Prototype.			
	Typically the accessibility and further information like alias names for a particular data is modeled on the level of DataPrototypes (especially VariableDataPrototypes, ParameterDataPrototypes). But due to the recursive structure of the meta-model concerning data types (a composite (data) type consists out of data prototypes) a part of the MCD information is described in the data type (in case of Application CompositeDataType).			
	This is a strong restriction in the reuse of data typed because the data type should be re-used for different VariableDataPrototypes and ParameterDataPrototypes to guarantee type compatibility on C-implementation level (e.g. data of a Port is stored in PIM or a ParameterDataPrototype used as ROM Block and shall be typed by the same data type as NVRAM Block).			
	This class overcomes such a restriction if applied properly.			
Base	ARObject			
Aggregated by	NvBlockDescriptor.instantiationDataDefProps, ParameterSwComponentType.instantiationDataDefProps, SwcInternalBehavior.instantiationDataDefProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
parameter Instance	AutosarParameterRef	0..1	aggr	This reference identifies the particular DataPrototype (defined in the context of a composite ParameterData Prototype) on which the swDataDefProps shall be applied.

▽

△

Class	InstantiationDataDefProps			
swDataDef Props	SwDataDefProps	0..1	aggr	These are the particular data definition properties which shall be applied **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=swDataDefProps
variableInstance	AutosarVariableRef	0..1	aggr	This reference identifies the particular DataPrototype (defined in the context of a composite VariableData Prototype) on which the swDataDefProps shall be applied.

Table A.560: InstantiationDataDefProps

Class	InstantiationRTEEventProps (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Composition			
Note	This meta-class represents the ability to refine the properties of RTEEvents for particular instances of a software component.			
Base	ARObject			
Subclasses	InstantiationTimingEventProps			
Aggregated by	CompositionSwComponentType.instantiationRTEEventProps			
Attribute	Type	Mult.	Kind	Note
refinedEvent	RTEEvent	0..1	iref	This instance ref denotes the Timing Event for which the period shall be refined on an instance level. **InstanceRef implemented by:**InstanceEventIn CompositionInstanceRef
shortLabel	Identifier	0..1	attr	The main purpose of the shortLabel is to contribute to the splitkey of aggregations that are <<atpSplitable>>. **Stereotypes:** atpIdentityContributor

Table A.561: InstantiationRTEEventProps

Class	InstantiationTimingEventProps			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Composition			
Note	This meta-class represents the ability to refine a timing event for particular instances of a software component. This approach supports an instance specific timing.			
Base	ARObject, InstantiationRTEEventProps			
Aggregated by	CompositionSwComponentType.instantiationRTEEventProps			
Attribute	Type	Mult.	Kind	Note
period	TimeValue	0..1	attr	This attribute represents the value of the refined activation period.

Table A.562: InstantiationTimingEventProps

Class	InternalBehavior (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	ARObject, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable, MultilanguageReferrable, Referrable
Subclasses	BswInternalBehavior, SwcInternalBehavior

▽

△

Class	*InternalBehavior* (abstract)			
Aggregated by	*AtpClassifier*.atpFeature			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
constant Memory	ParameterData Prototype	*	aggr	Describes a read only memory object containing characteristic value(s) implemented by this Internal Behavior.
				The shortName of ParameterDataPrototype has to be equal to the "C' identifier of the described constant.
				The characteristic value(s) might be shared between Sw ComponentPrototypes of the same SwComponentType.
				The aggregation of constantMemory is subject to variability with the purpose to support variability in the software component or module implementations. Typically different algorithms in the implementation are requiring different number of memory objects.
				Stereotypes: atpSplitable; atpVariation **Tags:** atp.Splitkey=constantMemory.shortName, constant Memory.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
constantValue Mapping	ConstantSpecification MappingSet	*	ref	Reference to the ConstantSpecificationMapping to be applied for the particular InternalBehavior
				Stereotypes: atpSplitable **Tags:**atp.Splitkey=constantValueMapping
dataType Mapping	DataTypeMappingSet	*	ref	Reference to the DataTypeMapping to be applied for the particular InternalBehavior
				Stereotypes: atpSplitable **Tags:**atp.Splitkey=dataTypeMapping
exclusiveArea	ExclusiveArea	*	aggr	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.
				Stereotypes: atpSplitable; atpVariation **Tags:** atp.Splitkey=exclusiveArea.shortName, exclusive Area.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
exclusiveArea NestingOrder	ExclusiveAreaNesting Order	*	aggr	This represents the set of ExclusiveAreaNestingOrder owned by the InternalBehavior.
				Stereotypes: atpSplitable; atpVariation **Tags:** atp.Splitkey=exclusiveAreaNestingOrder.shortName, exclusiveAreaNestingOrder.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	InternalBehavior (abstract)			
staticMemory	VariableDataPrototype	*	aggr	Describes a read and writeable static memory object representing measurerment 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. The shortName of the VariableDataPrototype has to be equal with the "C' identifier of the described variable. The aggregation of staticMemory 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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=staticMemory.shortName, static Memory.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table A.563: InternalBehavior

Class	InternalTriggerOccurredEvent			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents			
Note	This event is raised when the referenced InternalTriggeringPoint has occurred.			
Base	ARObject, AbstractEvent, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable, Multilanguage Referrable, RTEEvent, Referrable			
Aggregated by	AtpClassifier.atpFeature, SwcInternalBehavior.event			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
eventSource	InternalTriggeringPoint	0..1	ref	The referenced InternalTriggeringPoint raises this Internal TriggerOccurredEvent.

Table A.564: InternalTriggerOccurredEvent

Class	InternalTriggeringPoint			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::Trigger			
Note	If a RunnableEntity owns an InternalTriggeringPoint it is entitled to trigger the execution of Runnable Entities of the corresponding software-component.			
Base	ARObject, AbstractAccessPoint, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	AtpClassifier.atpFeature, RunnableEntity.internalTriggeringPoint			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
swImplPolicy	SwImplPolicyEnum	0..1	attr	This attribute, when set to value queued, allows for a queued processing of Triggers.

Table A.565: InternalTriggeringPoint

Class	InterpolationRoutine			
Package	M2::AUTOSARTemplates::SWComponentTemplate::MeasurementAndCalibration::InterpolationRoutineMappingSet			
Note	This represents an interpolation routine taken to evaluate the contents of a curve or map against a specific input value.			
Base	ARObject			
Aggregated by	InterpolationRoutineMapping.interpolationRoutine			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
interpolation Routine	BswModuleEntry	0..1	ref	This specifies a BswModuleEntry which implements the current interpolation method for the given record layout. **Tags:**xml.sequenceOffset=30
isDefault	Boolean	0..1	attr	This attribute specifies whether the enclosing InterpolationRoutine is considered the default in the context (defined by the System Template) of a given collection InterpolationRoutineMapping that owns the enclosing InterpolationRoutine. **Tags:**xml.sequenceOffset=20
shortLabel	Identifier	0..1	attr	This is the name of the interpolation method which is implemented by the referenced bswModuleEntry. It corresponds to swInterpolationMethod in SwDataDef Props. **Tags:**xml.sequenceOffset=10

Table A.566: InterpolationRoutine

Class	InterpolationRoutineMapping			
Package	M2::AUTOSARTemplates::SWComponentTemplate::MeasurementAndCalibration::InterpolationRoutineMappingSet			
Note	This meta-class provides a mapping between one record layout and its matching interpolation routines. This allows to formally specify the semantics of the interpolation routines. The use case is such that the curves/Maps define an interpolation method. This mapping table specifies which interpolation routine implements methods for a particular record layout. Using this information, the implementer of a software-component can select the appropriate interpolation routine.			
Base	ARObject			
Aggregated by	InterpolationRoutineMappingSet.interpolationRoutineMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
interpolation Routine	InterpolationRoutine	*	aggr	This is one particular interpolation routine which is mapped to the record layout.
swRecord Layout	SwRecordLayout	0..1	ref	This refers to the record layout which is mapped to interpolation routines.

Table A.567: InterpolationRoutineMapping

Class	InterpolationRoutineMappingSet
Package	M2::AUTOSARTemplates::SWComponentTemplate::MeasurementAndCalibration::InterpolationRoutineMappingSet
Note	This meta-class specifies a set of interpolation routine mappings. **Tags:**atp.recommendedPackage=InterpolationRoutineMappingSets
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*
Aggregated by	ARPackage.element

▽

△

Class	InterpolationRoutineMappingSet			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
interpolation Routine Mapping	InterpolationRoutine Mapping	*	aggr	This specifies one particular mapping of recordlayout and its matching interpolationRoutines.

Table A.568: InterpolationRoutineMappingSet

Class	InvalidationPolicy			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	Specifies whether the component can actively invalidate a particular dataElement. If no invalidationPolicy points to a dataElement this is considered to yield the identical result as if the handleInvalid attribute was set to dontInvalidate.			
Base	ARObject			
Aggregated by	SenderReceiverInterface.invalidationPolicy			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dataElement	VariableDataPrototype	0..1	ref	Reference to the dataElement for which the Invalidation Policy applies.
handleInvalid	HandleInvalidEnum	0..1	attr	This attribute controls how invalidation is applied to the dataElement.

Table A.569: InvalidationPolicy

Class	Ipv4ArpProps			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Specifies the configuration options for the ARP (Address Resolution Protocol).			
Base	ARObject			
Aggregated by	Ipv4Props.arpProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
tcpIpArpNum GratuitousArp OnStartup	PositiveInteger	0..1	attr	This attribute specifies the number of gratuitous ARP replies which shall be sent on assignment of a new IP address.
tcpIpArpPacket QueueEnabled	Boolean	0..1	attr	This attribute enables (TRUE) or disables (FALSE) support of the ARP Packet Queue according to IETF RFC 1122, section 2.3.2.2.
tcpIpArp Request Timeout	TimeValue	0..1	attr	This attribute specifies a timeout in seconds for the validity of ARP requests. After the transmission of an ARP request the TcpIp shall skip the transmission of any further ARP requests to the same destination within a duration of tcpIpArpRequestTimeout seconds. (IETF RFC 1122, section 2.3.2.1).
tcpIpArpTable EntryTimeout	TimeValue	0..1	attr	This attribute specifies the timeout in seconds after which an unused ARP entry is removed.

Table A.570: Ipv4ArpProps

Class	Ipv4Configuration	
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology	
Note	Internet Protocol version 4 (IPv4) configuration.	
Base	ARObject, NetworkEndpointAddress	

▽

△

Class	Ipv4Configuration			
Aggregated by	NetworkEndpoint.networkEndpointAddress			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
assignment Priority	PositiveInteger	0..1	attr	Priority of assignment (1 is highest). If a new address from an assignment method with a higher priority is available, it overwrites the IP address previously assigned by an assignment method with a lower priority.
defaultGateway	Ip4AddressString	0..1	attr	IP address of the default gateway.
dnsServer Address	Ip4AddressString	*	attr	IP addresses of preconfigured DNS servers. **Tags:**xml.namePlural=DNS-SERVER-ADDRESSES
ipAddressKeep Behavior	IpAddressKeepEnum	0..1	attr	Defines the lifetime of a dynamically fetched IP address.
ipv4Address	Ip4AddressString	0..1	attr	IPv4 Address. Notation: 255.255.255.255. The IP Address shall be declared in case the ipv4AddressSource is FIXED and thus no auto-configuration mechanism is used.
ipv4Address Source	Ipv4AddressSource Enum	0..1	attr	Defines how the node obtains its IP address.
networkMask	Ip4AddressString	0..1	attr	Network mask. Notation 255.255.255.255
ttl	PositiveInteger	0..1	attr	Lifespan of data (0..255). The purpose of the TimeToLive field is to avoid a situation in which an undeliverable datagram keeps circulating on a system.

Table A.571: Ipv4Configuration

Class	Ipv4FragmentationProps			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Specifies the configuration options for IPv4 packet fragmentation/reassembly.			
Base	ARObject			
Aggregated by	Ipv4Props.fragmentationProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
tcpIpIp Fragmentation RxEnabled	Boolean	0..1	attr	Enables (TRUE) or disables (FALSE) support for reassembling of incoming datagrams that are fragmented according to IETF RFC 815 (IP Datagram Reassembly Algorithms).
tcpIpIpNum Fragments	PositiveInteger	0..1	attr	Specifies the maximum number of IP fragments per datagram.
tcpIpIpNum ReassDgrams	PositiveInteger	0..1	attr	Specifies the maximum number of fragmented IP datagrams that can be reassembled in parallel.
tcpIpIpReass Timeout	TimeValue	0..1	attr	Specifies the timeout in [s] after which an incomplete datagram gets discarded.

Table A.572: Ipv4FragmentationProps

Class	Ipv6Configuration			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Internet Protocol version 6 (IPv6) configuration.			
Base	ARObject, NetworkEndpointAddress			
Aggregated by	NetworkEndpoint.networkEndpointAddress			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	Ipv6Configuration			
assignment Priority	PositiveInteger	0..1	attr	Priority of assignment (1 is highest). If a new address from an assignment method with a higher priority is available, it overwrites the IP address previously assigned by an assignment method with a lower priority.
defaultRouter	Ip6AddressString	0..1	attr	IP address of the default router.
dnsServer Address	Ip6AddressString	*	attr	IP addresses of pre configured DNS servers. **Tags:**xml.namePlural=DNS-SERVER-ADDRESSES
enableAnycast	Boolean	0..1	attr	This attribute is used to enable anycast addressing (i.e. to one of multiple receivers).
hopCount	PositiveInteger	0..1	attr	The distance between two hosts. The hop count n means that n gateways separate the source host from the destination host (Range 0..255)
ipAddressKeep Behavior	IpAddressKeepEnum	0..1	attr	Defines the lifetime of a dynamically fetched IP address.
ipAddressPrefix Length	PositiveInteger	0..1	attr	IPv6 prefix length defines the part of the IPv6 address that is the network prefix.
ipv6Address	Ip6AddressString	0..1	attr	IPv6 Address. Notation: FFFF:...:FFFF. The IP Address shall be declared in case the ipv6AddressSource is FIXED and thus no auto-configuration mechanism is used.
ipv6Address Source	Ipv6AddressSource Enum	0..1	attr	Defines how the node obtains its IP address.

Table A.573: Ipv6Configuration

Class	Ipv6FragmentationProps			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	This meta-class specifies the configuration options for IPv6 packet fragmentation/reassembly.			
Base	ARObject			
Aggregated by	Ipv6Props.fragmentationProps			
Attribute	Type	Mult.	Kind	Note
tcpIpIp Reassembly BufferCount	PositiveInteger	0..1	attr	Number of buffers that can be used for fragment reassembly. In case of a reassembly error or if not all fragments are received in time this buffer will be blocked until the specified "Fragment Reassembly Timeout" has been exceeded. A value of 0 disables fragment reassembly.
tcpIpIp Reassembly BufferSize	PositiveInteger	0..1	attr	Size of each fragment tx buffer in bytes.
tcpIpIp Reassembly SegmentCount	PositiveInteger	0..1	attr	Specifies the maximum number of consecutive data segments that can be managed in each reassembly buffer. If all fragments are received in order, only one segment will be needed. To deal with fragments received out of order this value should be configured bigger than 1.
tcpIpIp Reassembly Timeout	TimeValue	0..1	attr	Specifies the timeout in seconds after which an incomplete datagram gets discarded.

▽

△

Class	Ipv6FragmentationProps			
tcpIpIpTx FragmentBuffer Count	PositiveInteger	0..1	attr	These buffers will be used if the IpV6 receives packets from the upper layer that do not fit into the MTU and thus must be fragmented. A value of 0 disables tx fragmentation.
tcpIpIpTx FragmentBuffer Size	PositiveInteger	0..1	attr	Size of each fragment tx buffer in bytes.

Table A.574: Ipv6FragmentationProps

Class	Ipv6NdpProps			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	This meta-class specifies the configuration options for the Neighbor Discovery Protocol for IPv6.			
Base	ARObject			
Aggregated by	Ipv6Props.ndpProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
tcpIpNdpDefault ReachableTime	TimeValue	0..1	attr	Configuration of the ReachableTime (s) specified in [RFC4861 6.3.2. Host Variables].
tcpIpNdpDefault RetransTimer	TimeValue	1	attr	Configures the default value (s) for the RetransTimer variable specified in [RFC4861 6.3.2. Host Variables].
tcpIpNdpDefault RouterListSize	PositiveInteger	0..1	attr	Maximum number of default router entries.
tcpIpNdp Defensive Processing	Boolean	0..1	attr	If enabled the NDP shall only process Neighbor Advertisements which are received in reaction to a previously transmitted Neighbor Solicitation as well as skipping updates to the Neighbor Cache based on received Neighbor Solicitations. If disabled all Neighbor Advertisements and Solicitations shall be processed as specified in RFC4861.
tcpIpNdpDelay FirstProbeTime	PositiveInteger	0..1	attr	Delay before sending the first NUD probe in (s).
tcpIpNdp Destination CacheSize	PositiveInteger	0..1	attr	Maximum number of entries in the destination cache.
tcpIpNdp DynamicHop LimitEnabled	Boolean	0..1	attr	If enabled the default hop limit may be reconfigured based on received Router Advertisements.
tcpIpNdp DynamicMtu Enabled	Boolean	0..1	attr	Allow dynamic reconfiguration of link MTU via Router Advertisements.
tcpIpNdp Dynamic ReachableTime Enabled	Boolean	0..1	attr	If enabled the default Reachable Time value may be reconfigured based on received Router Advertisements.
tcpIpNdp Dynamic RetransTime Enabled	Boolean	0..1	attr	If enabled the default Retransmit Timer value may be reconfigured based on received Router Advertisements.
tcpIpNdpMax RandomFactor	PositiveInteger	0..1	attr	Maximum random factor used for randomization
tcpIpNdpMaxRtr Solicitation Delay	TimeValue	0..1	attr	Maximum delay before the first Router Solicitation will be sent after interface initialization in (s).

▽

△

Class	**Ipv6NdpProps**			
tcpIpNdpMaxRtr Solicitations	PositiveInteger	0..1	attr	Maximum number of Router Solicitations that will be sent before the first Router Advertisement has been received.
tcpIpNdpMin RandomFactor	PositiveInteger	0..1	attr	Minimum random factor used for randomization
tcpIpNdp Neighbor Unreachability Detection Enabled	Boolean	0..1	attr	Neighbor Unreachability Detection is used to remove unused entries from the neighbor cache. This feature is a basic feature of NDP and should be turned on.
tcpIpNdpNum Multicast Solicitations	PositiveInteger	0..1	attr	Maximum number of multicast solicitations that will be sent when performing address resolution.
tcpIpNdpNum Unicast Solicitations	PositiveInteger	0..1	attr	Maximum number of unicast solicitations that will be sent when performig Neighbor Unreachability Detection.
tcpIpNdpPacket QueueEnabled	Boolean	0..1	attr	Enables (TRUE) or disables (FALSE) support of a NDP Packet Queue according to IETF RFC 4861, section 7.2.2.
tcpIpNdpPrefix ListSize	PositiveInteger	0..1	attr	Maximum number of entries in the on-link prefix list.
tcpIpNdp Random ReachableTime Enabled	Boolean	0..1	attr	If enabled the value of ReachableTime will be multiplied with a random value between MIN_RANDOM_FACTOR and MAX_RANDOM_FACTOR in order to prevent multiple nodes from transmitting at exactly the same time.
tcpIpNdpRndRtr Solicitation DelayEnabled	Boolean	0..1	attr	If enabled the first router solicitation will be delayed randomly from [0...MAX_RTR_SOLICITATION_DELAY]. Otherwise the first router solicitation will be sent after exactly MAX_RTR_SOLICITATION_DELAY milliseconds.
tcpIpNdpRtr Solicitation Interval	TimeValue	0..1	attr	Interval between consecutive Router Solicitations in (s).
tcpIpNdpSlaac DadNumberOf Transmissions	PositiveInteger	0..1	attr	Number of Neighbor Solicitations that have to be unanswered in order to set an autoconfigurated address to PREFERRED (usable) state.
tcpIpNdpSlaac Dad Retransmission Delay	TimeValue	0..1	attr	Sets the maximum value for the address configuration delay (s).
tcpIpNdpSlaac DelayEnabled	Boolean	0..1	attr	If enabled transmission of the first DAD Neighbor Solicitation will be delayed by a random value from [0...MAX_DAD_DELAY].
tcpIpNdpSlaac OptimisticDad Enabled	Boolean	0..1	attr	Enable Optimistic Duplicate Address Detection (DAD) according to RFC4429.

Table A.575: Ipv6NdpProps

Class	<<atpVariation>> **J1939Cluster**
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology
Note	J1939 specific cluster attributes.
	Tags:atp.recommendedPackage=CommunicationClusters
Base	*ARObject*, *AbstractCanCluster*, *CollectableElement*, *CommunicationCluster*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*
Aggregated by	ARPackage.element

▽

△

Class	<<atpVariation>> **J1939Cluster**			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
networkId	PositiveInteger	0..1	attr	This represents the network ID for the J1939 cluster.
request2Support	Boolean	0..1	attr	Enables support for the Request2 PGN (RQST2).
usesAddress Arbitration	Boolean	0..1	attr	Defines whether the nodes attached to this channel use an initial address claim, and whether they react to contending address claims of other nodes. True: The initial address claim is sent, and the node reacts to address claims of other nodes. False: The node only sends an address claim upon request, and does not care for contending address claims.

Table A.576: J1939Cluster

Class	**J1939ControllerApplication**			
Package	M2::AUTOSARTemplates::SystemTemplate::SWmapping			
Note	This element represents a J1939 controller application. **Tags:**atp.recommendedPackage=J1939ControllerApplications			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
functionId	PositiveInteger	1	attr	This attribute represents the numerical function id of the J1939 controller application.
swComponent Prototype	SwComponent Prototype	0..1	iref	This represents the SwComponentPrototype (which is typically typed by a CompositionSwComponentType) that corresponds to the J1939ControllerApplication. **InstanceRef implemented by:**ComponentInSystem InstanceRef

Table A.577: J1939ControllerApplication

Class	**J1939ControllerApplicationToJ1939NmNodeMapping**			
Package	M2::AUTOSARTemplates::SystemTemplate::SWmapping			
Note	This meta-class represents the ability to map a J1939ControllerApplication to a J1939NmNode. Note that this is similar but not identical to the mapping of SwComponentPrototypes to EcuInstances; for J1939 the semantics of an EcuInstance itself is basically replaced by a J1939NmNode.			
Base	*ARObject*			
Aggregated by	SystemMapping.j1939ControllerApplicationToJ1939NmNodeMapping			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
j1939Controller Application	J1939Controller Application	0..1	ref	Reference to the J1939 Controller Application that is mapped to the referenced J1939NmNode.
j1939NmNode	J1939NmNode	0..1	ref	J1939NmNode that is the target of the J1939Controller ApplicationTo1939NmNodeMapping.

Table A.578: J1939ControllerApplicationToJ1939NmNodeMapping

Class	J1939DcmIPdu
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication
Note	Represents the IPdus handled by J1939Dcm. **Tags:**atp.recommendedPackage=Pdus
Base	*ARObject*, *CollectableElement*, *FibexElement*, *IPdu*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Pdu*, *Referrable*
Aggregated by	ARPackage.element

Attribute	Type	Mult.	Kind	Note
diagnostic MessageType	PositiveInteger	0..1	attr	This attribute is used to identify the actual DMx message, e.g 1 means DM01, etc.

Table A.579: J1939DcmIPdu

Class	J1939NmCluster
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement
Note	J1939 specific NmCluster attributes
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *NmCluster*, *Referrable*
Aggregated by	NmConfig.nmCluster

Attribute	Type	Mult.	Kind	Note
addressClaim Enabled	Boolean	0..1	attr	This attribute specifies whether the J1939Nm Bsw module is used or not. If this attribute is set to false then the J1939Nm configuration shall not be derived from the system description. But even in this case the nmNodeId might still be necessary for the J1939Rm and J1939Tp.

Table A.580: J1939NmCluster

Class	J1939NmNode
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement
Note	J1939 specific NM Node attributes.
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *NmNode*, *Referrable*
Aggregated by	*NmCluster*.nmNode

Attribute	Type	Mult.	Kind	Note
nodeName	J1939NodeName	0..1	aggr	NodeName configuration

Table A.581: J1939NmNode

Class	J1939NodeName
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement
Note	This element contains attributes to configure the J1939NmNode NAME.
Base	*ARObject*
Aggregated by	J1939NmNode.nodeName

Attribute	Type	Mult.	Kind	Note
arbitrary Address Capable	Boolean	1	attr	Arbitrary Address Capable field of the NAME of this node.
ecuInstance	Integer	1	attr	ECU Instance field of the NAME of this node.
function	Integer	1	attr	Function field of the NAME of this node.
function Instance	Integer	1	attr	Function Instance field of the NAME of this node.

▽

Document ID 635: AUTOSAR_TR_ClassicAutosarModelConstraints

△

Class	J1939NodeName			
identitiyNumber	Integer	1	attr	Identity Number field of the NAME of this node.
industryGroup	Integer	1	attr	Industry Group field of the NAME of this node.
manufacturer Code	Integer	1	attr	Manufacturer Code field of the NAME of this node.
vehicleSystem	Integer	1	attr	Vehicle System field of the NAME of this node.
vehicleSystem Instance	Integer	1	attr	Vehicle System Instance field of the NAME of this node.

Table A.582: J1939NodeName

Class	J1939TpPg			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	A J1939TpPg represents one J1939 message (parameter group, PG) identified by the PGN (parameter group number) that can be received or transmitted via J1939Tp.			
Base	ARObject			
Aggregated by	J1939TpConnection.tpPg			
Attribute	Type	Mult.	Kind	Note
directPdu	NPdu	0..1	ref	In case of variable length IPdus (with system signals of variable length), an additional NPdu (with the PGN in the CAN ID) is used for messages with up to 8 bytes.
pgn	Integer	0..1	attr	Parameter group number (PGN) of a J1939 message (parameter group, PG) that can be received or transmitted via J1939Tp. The PGN may be omitted when the a directPdu is referenced and is mapped into a Can FrameTriggering with an identifier.
requestable	Boolean	0..1	attr	Parameter Group can be triggered by the J1939 request message.
sdu	IPdu	*	ref	Reference to IPdus that are segmented by the Transport Protocol. If more than one IPdu is referenced, the IPdus are used when the same PGN is received in parallel via different transport protocols (BAM, CMDT, direct) on the same J1939TpConnection.

Table A.583: J1939TpPg

Class	<<atpMixedString>> LParagraph			
Package	M2::MSR::Documentation::TextModel::LanguageDataModel			
Note	This is the text for a paragraph in one particular language. The language is denoted in the attribute l.			
Base	ARObject, LanguageSpecific, MixedContentForParagraph			
Aggregated by	MultiLanguageParagraph.l1			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.584: LParagraph

Class	LatencyTimingConstraint			
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint::LatencyTimingConstraint			
Note	This constraint type restricts the time duration between the occurrence of the stimulus and the occurrence of the corresponding response of that chain. Two latency constraint types are of interest for automotive systems. These are the age of a certain response and the reaction to a certain stimulus. In contrast to OffsetTimingConstraint, a causal dependency between the stimulus and response event of the associated event chain is required.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TimingConstraint*, *Traceable*			
Aggregated by	*TimingExtension*.timingGuarantee, *TimingExtension*.timingRequirement			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
latency ConstraintType	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 A.585: LatencyTimingConstraint

Class	LifeCycleInfo			
Package	M2::AUTOSARTemplates::GenericStructure::LifeCycles			
Note	LifeCycleInfo describes the life cycle state of an element together with additional information like what to use instead			
Base	*ARObject*			
Aggregated by	LifeCycleInfoSet.lifeCycleInfo			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
lcObject	Referrable	1	ref	Element(s) have the life cycle as described in lcState.
lcState	LifeCycleState	0..1	ref	This denotes the particular state assigned to the object. If no lcState is given then the default life cycle state of Life CycleInfoSet is assumed.
periodBegin	LifeCyclePeriod	0..1	aggr	Starting point of period in which the element has the denoted life cycle state lcState. If no periodBegin is given then the default period begin of LifeCycleInfoSet is assumed.
periodEnd	LifeCyclePeriod	0..1	aggr	Expiry date, i.e. end point of period the element does not have the denoted life cycle state lcState any more. If no periodEnd is given then the default period begin of Life CycleInfoSet is assumed.
remark	DocumentationBlock	0..1	aggr	Remark describing for example • why the element was given the specified life cycle • the semantics of useInstead

$\bigtriangledown$

△

Class	LifeCycleInfo				
useInstead	Referrable	*	ref	Element(s) that should be used instead of the one denoted in referrable.
				Only relevant in case of life cycle states lcState unlike "valid". In case there are multiple references the exact semantics shall be individually described in the remark.

Table A.586: LifeCycleInfo

Class	LifeCycleInfoSet				
Package	M2::AUTOSARTemplates::GenericStructure::LifeCycles				
Note	This meta class represents the ability to attach a life cycle information to a particular set of elements.				
	The information can be defined for a particular period. This supports the definition of transition plans.				
	If no period is specified, the life cycle state applies forever.				
	Tags:atp.recommendedPackage=LifeCycleInfoSets				
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*				
Aggregated by	ARPackage.element				
Attribute	**Type**	**Mult.**	**Kind**	**Note**	
defaultLcState	LifeCycleState	1	ref	This denotes the default life cycle state. To be used in all LifeCycleInfo elements within the LifeCycleInfoSet if no life cycle state is stated there explicitly. I.e. the defaultLcState can be overwritten in LifeCycleInfo elements.	
defaultPeriodBegin	LifeCyclePeriod	0..1	aggr	Default starting point of period in which all the specified lifeCycleInfo apply. Note that the default period can be overridden for each lifeCycleInfo individually.	
defaultPeriodEnd	LifeCyclePeriod	0..1	aggr	Default expiry date, i.e. default end point of period for which all specified lifeCycleInfo apply. Note that the default period can be overridden for each lifeCycleInfo individually.	
lifeCycleInfo	LifeCycleInfo	*	aggr	This represents one particular life cycle information.	
usedLifeCycleStateDefinitionGroup	LifeCycleStateDefinitionGroup	1	ref	This denotes the life cycle states applicable to the current life cycle info set.	

Table A.587: LifeCycleInfoSet

Class	LifeCyclePeriod				
Package	M2::AUTOSARTemplates::GenericStructure::LifeCycles				
Note	This meta class represents the ability to specify a point of time within a specified period, e.g. the starting or end point, in which a specific life cycle state is valid/applies to.				
Base	*ARObject*				
Aggregated by	LifeCycleInfo.periodBegin, LifeCycleInfo.periodEnd, LifeCycleInfoSet.defaultPeriodBegin, LifeCycleInfoSet.defaultPeriodEnd				
Attribute	**Type**	**Mult.**	**Kind**	**Note**	
arReleaseVersion	RevisionLabelString	0..1	attr	Version of the AUTOSAR Release the element referred to is part of.	
				The numbering contains three levels (major, minor, revision) which are defined by AUTOSAR.	
				Tags:xml.sequenceOffset=20	

▽

△

Class	LifeCyclePeriod			
date	DateTime	0..1	attr	Date within period. **Tags:**xml.sequenceOffset=10
productRelease	RevisionLabelString	0..1	attr	Version of the product within the period. **Tags:**xml.sequenceOffset=30

Table A.588: LifeCyclePeriod

Class	LifeCycleState			
Package	M2::AUTOSARTemplates::GenericStructure::LifeCycles			
Note	This meta class represents one particular state in the LifeCycle.			
Base	*ARObject*, *AtpBlueprint*, *AtpBlueprintable*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	LifeCycleStateDefinitionGroup.lcState			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.589: LifeCycleState

Class	LifeCycleStateDefinitionGroup			
Package	M2::AUTOSARTemplates::GenericStructure::LifeCycles			
Note	This meta class represents the ability to define the states and properties of one particular life cycle. **Tags:**atp.recommendedPackage=LifeCycleStateDefintionGroups			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpBlueprintable*, *CollectableElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
lcState	LifeCycleState	*	aggr	Describes a single life cycle state of this life cycle state definition group.

Table A.590: LifeCycleStateDefinitionGroup

Primitive	Limit			
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::PrimitiveTypes			
Note	This class represents the ability to express a numerical limit. Note that this is in fact a NumericalVariation Point but has the additional attribute intervalType. **Tags:** xml.xsd.customType=LIMIT-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]+)?)([eE]([+\-]?)[0-9]+)?)|\.0|INF|-INF|NaN xml.xsd.type=string			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
intervalType	IntervalTypeEnum	0..1	attr	This specifies the type of the interval. If the attribute is missing the interval shall be considered as "CLOSED". **Tags:**xml.attribute=true

Table A.591: Limit

Document ID 635: AUTOSAR_TR_ClassicAutosarModelConstraints

Class	LinCommunicationConnector			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinTopology			
Note	LIN bus specific communication connector attributes.			
Base	ARObject, CommunicationConnector, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	EcuInstance.connector, MachineDesign.communicationConnector			
Attribute	Type	Mult.	Kind	Note
initialNad	Integer	0..1	attr	Initial NAD of the LIN slave.
linConfigurable Frame	LinConfigurableFrame	*	aggr	LinConfigurableFrames shall list all frames (unconditional frames, event-triggered frames and sporadic frames) processed by the slave node. This element is necessary for the LIN 2.0 Assign-Frame command.
linOrdered Configurable Frame	LinOrderedConfigurable Frame	*	aggr	LinOrderedConfigurableFrames shall list all frames (unconditional frames, event-triggered frames and sporadic frames) processed by the slave node. This element is necessary for the LIN 2.1 Assign-Frame-PID-Range command.
schedule ChangeNext TimeBase	Boolean	0..1	attr	This attribute defines the point in time where a schedule table switch is performed. If this attribute is set to false or not present, the schedule table shall be switched after the current entry of the active schedule table is ended. If this attribute is enabled, the schedule table shall be switched when message transmission or reception within an entry has been completed, ensured by status checks for transmission and reception.

Table A.592: LinCommunicationConnector

Class	LinConfigurableFrame			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinTopology			
Note	Assignment of messageIds to Frames. This element shall be used for the LIN 2.0 Assign-Frame command.			
Base	ARObject			
Aggregated by	LinCommunicationConnector.linConfigurableFrame, LinSlaveConfig.linConfigurableFrame			
Attribute	Type	Mult.	Kind	Note
frame	LinFrame	1	ref	Reference to a Frame that is processed by the slave node.
messageId	PositiveInteger	0..1	attr	MessageId for the referenced frame

Table A.593: LinConfigurableFrame

Class	LinErrorResponse			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	Each slave node shall publish a one bit signal, named response_error, to the master node in one of its transmitted unconditional frames. The response_error signal shall be set whenever a frame (except for event triggered frame responses) that is transmitted or received by the slave node contains an error in the frame response. The response_error signal shall be cleared when the unconditional frame containing the response_error signal is successfully transmitted.			
Base	ARObject			
Aggregated by	LinSlave.linErrorResponse, LinSlaveConfig.linErrorResponse			
Attribute	Type	Mult.	Kind	Note
responseError	ISignalTriggering	0..1	ref	This ISignal shall be taken to transport the responseError bit.

Table A.594: LinErrorResponse

Class	LinEventTriggeredFrame			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	An event triggered frame is used as a placeholder to allow multiple slave nodes to provide its response.			
	The header of an event triggered frame is transmitted when a frame slot allocated to the event triggered frame is processed. The publisher of an associated unconditional frame shall only transmit the response if at least one of the signals carried in its unconditional frame is updated. The LIN Master discovers and purges collisions with the collisionResolvingScheduleTable.			
	The event controlled frame shall not contain any Pdus.			
	Tags:atp.recommendedPackage=Frames			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Frame*, *Identifiable*, *LinFrame*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
collision Resolving Schedule	LinScheduleTable	0..1	ref	Reference to the schedule table, which resolves a collision.
linUnconditional Frame	LinUnconditionalFrame	1..*	ref	A list of slaves can respond to the master request if at least one of the signals carried in its unconditional frame is updated. For each response a LinFrameTriggering and a LinUnconditionalFrame shall be defined. Within a channel a LIN Frame shall be referenced by only one FrameTriggering. This allows a derivation of the identifier of a substituted Frame. The identifier is specified in FrameTriggering element. The Unconditional frames associated with an event triggered frame shall: • have equal length. • use the same checksum model (i.e. mixing LIN 1.x and LIN 2.x frames is not allowed). • reserve the first data field to its protected identifier (even if the associated unconditional frame is scheduled as a unconditional frame in the same or another schedule table). • be published by different slave nodes. • shall not be included directly in the same schedule table as the event triggered frame is scheduled.

Table A.595: LinEventTriggeredFrame

Class	LinFrame (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	Lin specific Frame element.			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Frame*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	LinEventTriggeredFrame, LinSporadicFrame, LinUnconditionalFrame			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.596: LinFrame

Class	**LinFrameTriggering**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	LIN specific attributes to the FrameTriggering			
Base	*ARObject*, *FrameTriggering*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*PhysicalChannel*.frameTriggering			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
identifier	Integer	0..1	attr	To describe a frames identifier on the communication system, usually with a fixed identifierValue. For Lin SporadicFrames the attribute shall be ignored.
linChecksum	LinChecksumType	0..1	attr	Type of checksum that the frame is using. This attribute is optional because in case of sporadic frames it should not be set.

Table A.597: LinFrameTriggering

Class	<<atpVariation>> **LinMaster**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinTopology			
Note	Describing the properties of the refering ecu as a LIN master.			
Base	*ARObject*, *CommunicationController*, *Identifiable*, LinCommunicationController, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EcuInstance.commController, MachineDesign.communicationController			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
linSlave	LinSlaveConfig	*	aggr	LinSlaves that are handled by the LinMaster.
timeBase	TimeValue	0..1	attr	Time base is mandatory for the master. It is not used for slaves. LIN 2.0 Spec states: "The time_base value specifies the used time base in the master node to generate the maximum allowed frame transfer time." The time base shall be specified AUTOSAR conform in seconds.
timeBaseJitter	TimeValue	0..1	attr	The attribute timeBaseJitter is a mandatory attribute for the master and not used for slaves. LIN 2.0 Spec states: "The jitter value specifies the differences between the maximum and minimum delay from time base start point to the frame header sending start point (falling edge of BREAK signal)." The jitter shall be specified AUTOSAR conform in seconds.

Table A.598: LinMaster

Class	**LinOrderedConfigurableFrame**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinTopology			
Note	With the assignment of the index to a frame a mapping of Pids to Frames is possible. This element shall be used for the LIN 2.1 Assign-Frame-PID-Range command.			
Base	*ARObject*			
Aggregated by	LinCommunicationConnector.linOrderedConfigurableFrame, LinSlaveConfig.linOrderedConfigurableFrame			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
frame	LinFrame	1	ref	Reference to a Frame that is processed by the slave node.

▽

△

Class	LinOrderedConfigurableFrame			
index	Integer	1	attr	This attribute is used to order the elements and allows an assignment of Pids to ConfigurableFrames that are defined in the slave.

Table A.599: LinOrderedConfigurableFrame

Class	LinPhysicalChannel			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinTopology			
Note	LIN specific attributes to the physicalChannel			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *PhysicalChannel*, *Referrable*			
Aggregated by	*CommunicationCluster*.physicalChannel			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
busIdleTimeout Period	TimeValue	0..1	attr	This attribute shall be used to set an idle timeout period for the enclosing LinPhysicalChannel.
scheduleTable	LinScheduleTable	*	aggr	Schedule tables organize the timings of the frames for LIN. atpVariation: If the transmitted frames are variable, the corresponding ScheduleTables shall be variable, too. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=scheduleTable.shortName, schedule Table.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.600: LinPhysicalChannel

Class	LinScheduleTable			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	The master task (in the master node) transmits frame headers based on a schedule table. The schedule table specifies the identifiers for each header and the interval between the start of a frame and the start of the following frame.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	LinPhysicalChannel.scheduleTable			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
resumePosition	ResumePosition	0..1	attr	Defines, where a schedule table shall be proceeded in case if it has been interrupted by a run-once table or MRF/SRF.
runMode	RunMode	0..1	attr	The schedule table can be executed in two different modes.
tableEntry	ScheduleTableEntry	1..*	aggr	The scheduling table consists of table entries, which contain Frame slots.

Table A.601: LinScheduleTable

Class	<<atpVariation>> LinSlave
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinTopology
Note	Describing the properties of the referring ecu as a LIN slave.
Base	*ARObject*, *CommunicationController*, *Identifiable*, LinCommunicationController, *Multilanguage Referrable*, *Referrable*

▽

△

Class	<<atpVariation>> **LinSlave**			
Aggregated by	EcuInstance.commController, MachineDesign.communicationController			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
assignNad	Boolean	0..1	attr	This attribute has the ability to control whether the node configuration command 'Assign NAD' is supported.
configuredNad	Integer	1	attr	To distinguish LIN slaves that are used twice or more within the same cluster.
functionId	PositiveInteger	1	attr	LIN function ID
initialNad	Integer	0..1	attr	This attribute represents the initial NAD.
linError Response	LinErrorResponse	1	aggr	Each slave node shall publish one response error in one of its transmitted unconditional frames.
nasTimeout	TimeValue	0..1	attr	Value of the N_AS timeout. Unit: seconds.
supplierId	PositiveInteger	1	attr	LIN Supplier ID
variantId	PositiveInteger	1	attr	Specifies the Variant ID

Table A.602: LinSlave

Class	**LinSlaveConfig**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinTopology			
Note	Node attributes of LIN slaves that are handled by the LinMaster. In the System Description LIN slaves may be described in the context of the Lin Master. In an ECU Extract of the LinMaster the LinSlave Ecus shall not be available. The information that is described here is necessary in the ECU Extract for the configuration of the Lin Master. The values of attributes of LinSlaveConfig and the corresponding LinSlave shall be identical (if both are defined in a System Description).			
Base	ARObject			
Aggregated by	LinMaster.linSlave			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
configuredNad	Integer	1	attr	To distinguish LIN slaves that are used twice or more within the same cluster.
functionId	PositiveInteger	1	attr	LIN function ID.
ident	LinSlaveConfigIdent	0..1	aggr	This adds the ability to become referrable to LinSlave Config.
initialNad	Integer	0..1	attr	Initial NAD of the LIN slave.
linConfigurable Frame	LinConfigurableFrame	*	aggr	List of all frames that are processed by the slave node
linError Response	LinErrorResponse	0..1	aggr	Each slave node shall publish one response error in one of its transmitted unconditional frames.
linOrdered Configurable Frame	LinOrderedConfigurable Frame	*	aggr	List of all frames (unconditional frames, event-triggered frames and sporadic frames) processed by the slave node. This element is necessary for the LIN 2.1 Assign-Frame-PID-Range command.
protocolVersion	String	0..1	attr	Version specifier for a communication protocol. Protocol version of the LinMaster and the LinSlaves may be different.
supplierId	PositiveInteger	1	attr	LIN Supplier ID.
variantId	PositiveInteger	1	attr	Specifies the Variant ID.

Table A.603: LinSlaveConfig

Class	LinSporadicFrame			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	A sporadic frame is a group of unconditional frames that share the same frame slot. The sporadic frame shall not contain any Pdus. **Tags:**atp.recommendedPackage=Frames			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Frame*, *Identifiable*, *LinFrame*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
substituted Frame (ordered)	LinUnconditionalFrame	1..*	ref	Reference to a group of unconditional frames that share the same frame slot. In case that more than one of the declared frames needs to be transferred, the one first listed shall be chosen. Within a channel a LIN Frame shall be referenced by only one FrameTriggering. This allows a derivation of the identifier of a substituted Frame. The identifier is specified in FrameTriggering element. A LinUnconditionalFrame associated with a LinSporadic Frame may not be allocated in the same LinSchedule Table as the sporadic frame.

Table A.604: LinSporadicFrame

Class	LinUnconditionalFrame			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	Unconditional frames carry signals. The master sends a frame header in a scheduled frame slot and the designated slave node fills the frame with data. **Tags:**atp.recommendedPackage=Frames			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Frame*, *Identifiable*, *LinFrame*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.605: LinUnconditionalFrame

Class	MacMulticastGroup			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Per EthernetCluster globally defined MacMulticastGroup. One sender can handle many receivers simultaneously if the receivers have all the same macMulticastAddress. The addresses need to be unique for the particular EthernetCluster.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	EthernetCluster.macMulticastGroup			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
macMulticast Address	MacAddressString	1	attr	A multicast MAC address (Media Access Control address) is a identifier for a group of hosts in a network.

Table A.606: MacMulticastGroup

Class	MacSecProps			
Package	M2::AUTOSARTemplates::SystemTemplate::SecureCommunication			
Note	This meta-class allows to configure MACsec (Media access control security) and the MKA (MACsec Key Agreement) for the CouplingPort (PHY). **Tags:**atp.Status=candidate			
Base	ARObject			
Aggregated by	CouplingPort.macSecProps			
Attribute	Type	Mult.	Kind	Note
autoStart	Boolean	0..1	attr	This attribute defines how the Port Access Entity (PAE) is started: True := Autostart False := Manual Start **Tags:**atp.Status=candidate
macSecKay Config	MacSecLocalKayProps	0..1	aggr	Properties to configure the MKA instance (KaY) for a controlled CouplingPort (PaE). **Tags:**atp.Status=candidate
onFail Permissive Mode	MacSecFailPermissive ModeEnum	0..1	attr	This attribute sets the behavior of the Port Access Entity in case MACsec does not succeed. **Tags:**atp.Status=candidate
onFail Permissive ModeTimeout	TimeValue	0..1	attr	Timeout in seconds to enable the controlled port in case onFailPermissiveMode is set to Timeout. **Tags:**atp.Status=candidate
sakRekeyTime Span	TimeValue	0..1	attr	Time in seconds to trigger the rekey of an in use SAK (Static Secure Association key). If set to 0, the rekey will not be triggered after a time span. **Tags:**atp.Status=candidate

Table A.607: MacSecProps

Class	McDataAccessDetails			
Package	M2::AUTOSARTemplates::CommonStructure::MeasurementCalibrationSupport			
Note	This meta-class allows to attach detailed information about the usage of a data buffer by the RTE to a corresponding McDataInstance. Use Case: Direct memory access to RTE internal buffers for rapid prototyping. In case of implicit communication, the various task local buffers need to be identified in relation to RTE events and variable access points. Note that the SwComponentPrototype, the RunnableEntity and the VariableDataPrototype are implicitly given be the referred instances of RTEEvent and VariableAccess.			
Base	ARObject			
Aggregated by	McDataInstance.mcDataAccessDetails			
Attribute	Type	Mult.	Kind	Note
rteEvent	RTEEvent	*	iref	The RTE event used to receive the data via this buffer. **InstanceRef implemented by:**RteEventInEcuInstance Ref
variableAccess	VariableAccess	*	iref	The VariableAccess for which the data buffer is used. **InstanceRef implemented by:**VariableAccessInEcu InstanceRef

Table A.608: McDataAccessDetails

Class	McDataInstance
Package	M2::AUTOSARTemplates::CommonStructure::MeasurementCalibrationSupport
Note	Describes the specific properties of one data instance in order to support measurement and/or calibration of this data instance. The most important attributes are: • Its shortName is copied from the ECU Flat map (if applicable) and will be used as identifier and for display by the MC system. • The category is copied from the corresponding data type (ApplicationDataType if defined, otherwise ImplementationDataType) as far as applicable. • The symbol is the one used in the programming language. It will be used to find out the actual memory address by the final generation tool with the help of linker generated information. It is assumed that in the M1 model this part and all the aggregated and referred elements (with the exception of the Flat Map and the references from ImplementationElementInParameterInstanceRef and McAccessDetails) are completely generated from "upstream" information. This means, that even if an element like e.g. a CompuMethod is only used via reference here, it will be copied into the M1 artifact which holds the complete McSupportData for a given Implementation.
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*
Aggregated by	McDataInstance.subElement, McSupportData.mcParameterInstance, McSupportData.mcVariableInstance

Attribute	Type	Mult.	Kind	Note
arraySize	PositiveInteger	0..1	attr	The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of number of elements.
displayIdentifier	McdIdentifier	0..1	attr	An optional attribute to be used to set the ASAM ASAP2 DISPLAY_IDENTIFIER attribute.
flatMapEntry	FlatInstanceDescriptor	0..1	ref	Reference to the corresponding entry in the ECU Flat Map. This allows to trace back to the original specification of the generated data instance. This link shall be added by the RTE generator mainly for documentation purposes. The reference is optional because • The McDataInstance may represent an array or struct in which only the subElements correspond to FlatMap entries. • The McDataInstance may represent a task local buffer for rapid prototyping access which is different from the "main instance" used for measurement access.
instanceIn Memory	ImplementationElementInParameterInstanceRef	0..1	aggr	Reference to the corresponding data instance in the description of calibration data structures published by the RTE generator. This is used to support emulation methods inside the ECU, it is not required for A2L generation.
mcDataAccess Details	McDataAccessDetails	0..1	aggr	Refers to "upstream" information on how the RTE uses this data instance. Use Case: Rapid Prototyping
mcData Assignment	RoleBasedMcData Assignment	*	aggr	An assignment between McDataInstances. This supports the indication of related McDataElement implementing the of "RP global buffer", "RP global measurement buffer", "RP enabler flag".
resulting Properties	SwDataDefProps	0..1	aggr	These are the generated properties resulting from decisions taken by the RTE generator for the actually implemented data instance. Only those properties are relevant here, which are needed for the measurement and calibration system. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=resultingProperties

▽

△

Class	McDataInstance			
resultingRptSw Prototyping Access	RptSwPrototyping Access	0..1	aggr	Describes the implemented accessibility of data and modes by the rapid prototyping tooling.
role	Identifier	0..1	attr	An optional attribute to be used for additional information on the role of this data instance, for example in the context of rapid prototyping.
rptImplPolicy	RptImplPolicy	0..1	aggr	Describes the implemented code preparation for rapid prototyping at data accesses for a hook based bypassing.
subElement (ordered)	McDataInstance	*	aggr	This relation indicates, that the target element is part of a "struct" which is given by the source element. This information will be used by the final generator to set up the correct addressing scheme. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=subElement.shortName, sub Element.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
symbol	SymbolString	0..1	attr	This String is used to determine the memory address during final generation of the MC configuration data (e.g. "A2L" file) . It shall be the name of the element in the programming language such that it can be identified in linker generated information. In case the McDataInstance is part of composite data in the programming language, the symbol String may include parts denoting the element context, unless the context is given by the symbol attribute of an enclosing McDataInstance. This means in particular for the C language that the "." character shall be used as a separator between the name of a "struct" variable the name of one of its elements. The symbol can differ from the shortName in case of generated C data declarations. It is an optional attribute since it may be missing in case the instance represents an element (e.g. a single array element) which has no name in the linker map. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=symbol

Table A.609: McDataInstance

Class	McFunction			
Package	M2::AUTOSARTemplates::CommonStructure::MeasurementCalibrationSupport			
Note	Represents a functional element to be used as input to support measurement and calibration. It is used to • assign calibration parameters to a logical function • assign measurement variables to a logical function • structure functions hierarchically **Tags:**atp.recommendedPackage=McFunctions			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*

▽

△

Class	McFunction				
defCalprmSet	McFunctionDataRefSet	0..1	aggr		Refers to the set of adjustable data (= calibration parameters) defined in this function. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=defCalprmSet xml.sequenceOffset=10
inMeasurement Set	McFunctionDataRefSet	0..1	aggr		Refers to the set of measurable input data for this function. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=inMeasurementSet xml.sequenceOffset=30
loc Measurement Set	McFunctionDataRefSet	0..1	aggr		Refers to the set of measurable local data in this function. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=locMeasurementSet xml.sequenceOffset=50
out Measurement Set	McFunctionDataRefSet	0..1	aggr		Refers to the set of measurable output data from this function. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=outMeasurementSet xml.sequenceOffset=60
refCalprmSet	McFunctionDataRefSet	0..1	aggr		Refers to the set of adjustable data (= calibration parameters) referred by this function. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=refCalprmSet xml.sequenceOffset=20
subFunction	McFunction	*	ref		A sub-function that is seen as part of the enclosing function. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=subFunction xml.sequenceOffset=70

Table A.610: McFunction

Class	<<atpVariation>> **McFunctionDataRefSet**			
Package	M2::AUTOSARTemplates::CommonStructure::MeasurementCalibrationSupport::RptSupport			

Note	Refers to a set of data assigned to an McFunction in a particular role. The data are given
	
	    • either by entries in a FlatMap
	    • or by data instances that are part of MC support data.
	
	These two possibilities are exclusive within a given McFunctionDataRefSet. Which one to use depends on the process and tool environment.
	
	The set is subject to variability because the same functional model may be used with various representation of the data.
	
	Tags:vh.latestBindingTime=preCompileTime			

Base	ARObject			
Aggregated by	McFunction.defCalprmSet, McFunction.inMeasurementSet, McFunction.locMeasurementSet, Mc Function.outMeasurementSet, McFunction.refCalprmSet			
Attribute	*Type*	*Mult.*	*Kind*	*Note*

▽

△

Class	<<atpVariation>> **McFunctionDataRefSet**			
flatMapEntry	FlatInstanceDescriptor	*	ref	Refers to an entry in a FlatMap that is part of the set, for example a calibration parameter or measured variable.
				Note: This atpSplitable property has no atp.Splitkey due to atpVariation (PropertySetPattern).
				Stereotypes: atpSplitable **Tags:**xml.sequenceOffset=10
mcDataInstance	McDataInstance	*	ref	Refers to a data instance within MC support data that is part of the set, i.e. a calibration parameter or measured variable.
				Note: This atpSplitable property has no atp.Splitkey due to atpVariation (PropertySetPattern).
				Stereotypes: atpSplitable **Tags:**xml.sequenceOffset=20

Table A.611: McFunctionDataRefSet

Class	**McGroup**			
Package	M2::AUTOSARTemplates::CommonStructure::McGroups			
Note	Represents a group element to be used as input to support measurement and calibration. It is used to provide selection lists (groups) of calibration parameters, measurement variables, and functions in a hierarchical manner (subGroups).			
	Tags:atp.recommendedPackage=McFunctions			
Base	*ARElement*, ARObject, CollectableElement, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
mcFunction	McFunction	*	ref	A McFunction that is seen as part of the enclosing group.
				Stereotypes: atpSplitable **Tags:** atp.Splitkey=mcFunction xml.sequenceOffset=40
refCalprmSet	McGroupDataRefSet	0..1	aggr	Refers to the set of adjustable data (= calibration parameters) referred by this McGroup.
				Stereotypes: atpSplitable **Tags:** atp.Splitkey=refCalprmSet xml.sequenceOffset=20
ref Measurement Set	McGroupDataRefSet	0..1	aggr	Refers to the set of measurable belonging to this Mc Group.
				Stereotypes: atpSplitable **Tags:** atp.Splitkey=refMeasurementSet xml.sequenceOffset=30
subGroup	McGroup	*	ref	A sub-group that is seen as part of the enclosing group.
				Stereotypes: atpSplitable **Tags:** atp.Splitkey=subGroup xml.sequenceOffset=10

Table A.612: McGroup

Class	<<atpVariation>> **McGroupDataRefSet**			
Package	M2::AUTOSARTemplates::CommonStructure::McGroups			
Note	Refers to a set of data assigned to an McGroup in a particular role. The data are given • either by entries in a FlatMap • or by data instances that are part of MC support data. These two possibilities can be mixed within a given McGroupDataRefSet. Which one to use depends on the process and tool environment. The set is subject to variability because the same functional model may be used with various representation of the data. **Tags:**vh.latestBindingTime=preCompileTime			
Base	*ARObject*			
Aggregated by	McGroup.refCalprmSet, McGroup.refMeasurementSet			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
flatMapEntry	FlatInstanceDescriptor	*	ref	Refers to an entry in a FlatMap that is part of the set, for example a calibration parameter or measured variable. Note: This atpSplitable property has no atp.Splitkey due to atpVariation (PropertySetPattern). **Stereotypes:** atpSplitable **Tags:**xml.sequenceOffset=50
mcDataInstance	McDataInstance	*	ref	Refers to a data instance within MC support data that is part of the set, i.e. a calibration parameter or measured variable. Note: This atpSplitable property has no atp.Splitkey due to atpVariation (PropertySetPattern). **Stereotypes:** atpSplitable **Tags:**xml.sequenceOffset=60

Table A.613: McGroupDataRefSet

Class	**McParameterElementGroup**			
Package	M2::AUTOSARTemplates::CommonStructure::MeasurementCalibrationSupport			
Note	Denotes a group of calibration parameters which are handled by the RTE as one data structure.			
Base	*ARObject*			
Aggregated by	McSwEmulationMethodSupport.elementGroup			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
ramLocation	VariableDataPrototype	0..1	ref	Refers to the RAM location of this parameter group. To be used for the init-RAM method.
romLocation	ParameterData Prototype	0..1	ref	Refers to the ROM location of this parameter group. To be used for the init-RAM method.
shortLabel	Identifier	0..1	attr	Assigns a name to this element. **Tags:**xml.sequenceOffset=-100

Table A.614: McParameterElementGroup

Class	**McSupportData**
Package	M2::AUTOSARTemplates::CommonStructure::MeasurementCalibrationSupport
Note	Root element for all measurement and calibration support data related to one Implementation artifact on an ECU. There shall be one such element related to the RTE implementation (if it owns MC data) and a separate one for each module or component, which owns private MC data.
Base	*ARObject*

▽

$\triangle$

Class	McSupportData			
Aggregated by	*Implementation*.mcSupport			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
emulation Support	McSwEmulationMethod Support	*	aggr	Describes the calibration method used by the RTE. This information is not needed for A2L generation, but to setup software emulation in the ECU. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=emulationSupport, emulation Support.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
mcParameter Instance	McDataInstance	*	aggr	A data instance to be used for calibration. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=mcParameterInstance.shortName, mc ParameterInstance.variationPoint.shortLabel vh.latestBindingTime=postBuild
mcVariable Instance	McDataInstance	*	aggr	A data instance to be used for measurement. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=mcVariableInstance.shortName, mcVariable Instance.variationPoint.shortLabel vh.latestBindingTime=postBuild
measurable System ConstantValues	SwSystemconstant ValueSet	*	ref	Sets of system constant values to be transferred to the MCD system, because the system constants have been specified with "swCalibrationAccess" = readonly.
rptSupportData	RptSupportData	0..1	aggr	The rapid prototyping support data belonging to this implementation. The aggregtion is <<atpSplitable>> because in case of an already exisiting BSW Implementation model, this description will be added later in the process, namely at code generation time. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=rptSupportData

Table A.615: McSupportData

Class	McSwEmulationMethodSupport			
Package	M2::AUTOSARTemplates::CommonStructure::MeasurementCalibrationSupport			
Note	This denotes the method used by the RTE to handle the calibration data. It is published by the RTE generator and can be used e.g. to generate the corresponding emulation method in a Complex Driver. According to the actual method given by the category attribute, not all attributes are always needed: • double pointered method: only baseReference is mandatory • single pointered method: only referenceTable is mandatory • initRam method: only elementGroup(s) are mandatory Note: For single/double pointered method the group locations are implicitly accessed via the reference table and their location can be found from the initial values in the M1 model of the respective pointers. Therefore, the description of elementGroups is not needed in these cases. Likewise, for double pointered method the reference table description can be accessed via the M1 model under baseReference.			
Base	ARObject			
Aggregated by	McSupportData.emulationSupport			
Attribute	*Type*	*Mult.*	*Kind*	*Note*

$\triangledown$

△

Class	McSwEmulationMethodSupport			
baseReference	VariableDataPrototype	0..1	ref	Refers to the base pointer in case of the double-pointered method.
category	Identifier	0..1	attr	Identifies the actual method. The possible names shall correspond to the symbols of the ECU configuration parameter for the calibration method of the RTE, and can include vendor specific methods. **Tags:**xml.sequenceOffset=-90
elementGroup	McParameterElement Group	*	aggr	Denotes the grouping of calibration parameters in the actual RTE code. Depending on the category, this information maybe required to set up the emulation code.
referenceTable	VariableDataPrototype	0..1	ref	Refers to the pointer table in case of the single-pointered method.
shortLabel	Identifier	0..1	attr	Assigns a name to this element. **Tags:**xml.sequenceOffset=-100

Table A.616: McSwEmulationMethodSupport

Class	MeasuredExecutionTime			
Package	M2::AUTOSARTemplates::CommonStructure::ResourceConsumption::ExecutionTime			
Note	Specifies the ExecutionTime which has been gathered using measurement means.			
Base	ARObject, ExecutionTime, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	ResourceConsumption.executionTime			
Attribute	Type	Mult.	Kind	Note
maximum ExecutionTime	MultidimensionalTime	0..1	aggr	The maximum measured execution time.
minimum ExecutionTime	MultidimensionalTime	0..1	aggr	The minimum measured execution time.
nominal ExecutionTime	MultidimensionalTime	0..1	aggr	The nominal measured execution time.

Table A.617: MeasuredExecutionTime

Class	MeasuredHeapUsage			
Package	M2::AUTOSARTemplates::CommonStructure::ResourceConsumption::HeapUsage			
Note	The heap usage has been measured.			
Base	ARObject, HeapUsage, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	ResourceConsumption.heapUsage			
Attribute	Type	Mult.	Kind	Note
averageMemory Consumption	PositiveInteger	0..1	attr	The average heap usage measured. Unit: byte.
maximum Memory Consumption	PositiveInteger	0..1	attr	The maximum heap usage measured. Unit: byte.
minimum Memory Consumption	PositiveInteger	0..1	attr	The minimum heap usage measured. Unit: byte.
testPattern	String	0..1	attr	Description of the test pattern used to acquire the measured values.

Table A.618: MeasuredHeapUsage

Class	MeasuredStackUsage			
Package	M2::AUTOSARTemplates::CommonStructure::ResourceConsumption::StackUsage			
Note	The stack usage has been measured.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *StackUsage*			
Aggregated by	ResourceConsumption.stackUsage			
Attribute	Type	Mult.	Kind	Note
averageMemory Consumption	PositiveInteger	0..1	attr	The average stack usage measured. Unit: byte.
maximum Memory Consumption	PositiveInteger	0..1	attr	The maximum stack usage measured. Unit: byte.
minimum Memory Consumption	PositiveInteger	0..1	attr	The minimum stack usage measured. Unit: byte.
testPattern	String	0..1	attr	Description of the test pattern used to acquire the measured values.

Table A.619: MeasuredStackUsage

Enumeration	MemoryAllocationKeywordPolicyType
Package	M2::MSR::DataDictionary::AuxillaryObjects
Note	Enumeration to specify the name pattern of the Memory Allocation Keyword.
Aggregated by	SwAddrMethod.memoryAllocationKeywordPolicy
Literal	Description
addrMethodShort Name	The MemorySection shortNames of referring MemorySections and therefore the belonging Memory Allocation Keywords in the code are build with the shortName of the SwAddrMethod. This is the default value if the attribute does not exist. **Tags:**atp.EnumerationLiteralIndex=0
addrMethodShort NameAndAlignment	The MemorySection shortNames of referring MemorySections and therefore the belonging Memory Allocation Keywords in the code are build with the shortName of the SwAddrMethod and a variable alignment postfix. Thereby the alignment postfix needs to be consistent with the alignment attribute of the related MemorySection. **Tags:**atp.EnumerationLiteralIndex=1

Table A.620: MemoryAllocationKeywordPolicyType

Class	MemorySection			
Package	M2::AUTOSARTemplates::CommonStructure::ResourceConsumption::MemorySectionUsage			
Note	Provides a description of an abstract memory section used in the Implementation for code or data. It shall be declared by the Implementation Description of the module or component, which actually allocates the memory in its code. This means in case of data prototypes which are allocated by the RTE, that the generated Implementation Description of the RTE shall contain the corresponding MemorySections. The attribute "symbol" (if symbol is missing: "shortName") defines the module or component specific section name used in the code. For details see the document "Specification of Memory Mapping". Typically the section name is build according the pattern: <SwAddrMethod shortName>[_<further specialization nominator>][_<alignment>] where • **[<SwAddrMethod shortName>]** is the shortName of the referenced SwAddrMethod • **[_<further specialization nominator>]** is an optional infix to indicate the specialization in the case that several MemorySections for different purpose of the same Implementation Description referring to the same or equally named SwAddrMethods. • **[_<alignment>]** is the alignment attributes value and is only applicable in the case that the memoryAllocationKeywordPolicy value of the referenced SwAddrMethod is set to addrMethod ShortNameAndAlignment MemorySection used to Implement the code of RunnableEntitys and BswSchedulableEntitys shall have a symbol (if missing: shortName) identical to the referred SwAddrMethod to conform to the generated RTE header files. In addition to the section name described above, a prefix is used in the corresponding macro code in order to define a name space. This prefix is by default given by the shortName of the BswModule Description resp. the SwComponentType. It can be superseded by the prefix attribute.			
Base	_ARObject_, _Identifiable_, _MultilanguageReferrable_, _Referrable_			
Aggregated by	ResourceConsumption.memorySection			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
alignment	AlignmentType	0..1	attr	The attribute describes the typical alignment of objects within this memory section.
executableEntity	ExecutableEntity	*	ref	Reference to the ExecutableEntitites located in this section. This allows to locate different Executable Entitities in different sections even if the associated Sw Addrmethod is the same. This is applicable to code sections only.
option	Identifier	*	attr	The service (in AUTOSAR: BswModuleEntry) is implemented in a way that it either resolves to aninline function or to a standard function depending on conditions set at a later point in time. The following two values are standardized (to be used for code sections only and exclusively to each other): • INLINE - The code section is declared with the keyword "inline". • LOCAL_INLINE - The code section is declared with the keyword "static inline". In both cases (INLINE and LOCAL_INLINE) the inline expansion depends on the compiler. Depending on this, the code section either corresponds to an actual section in memory or is put into the section of the caller.
prefix	SectionNamePrefix	0..1	ref	The prefix used to set the memory section's namespace in the code. The existence of a prefix element supersedes rules for a default prefix (such as the Bsw ModuleDescription's shortName). This allows the user to define several name spaces for memory sections within the scope of one module, cluster or SWC.
size	PositiveInteger	0..1	attr	The size in bytes of the section.

▽

△

Class	MemorySection			
swAddrmethod	SwAddrMethod	0..1	ref	This association indicates that this module specific (abstract) memory section is part of an overall SwAddr Method, referred by the upstream declarations (e.g. calibration parameters, data element prototypes, code entities) which share a common addressing strategy. This can be evaluated for the ECU configuration of the build support. This association shall always be declared by the Implementation description of the module or component, which allocates the memory in its code. This means in case of data prototypes which are allocated by the RTE, that the software components only declare the grouping of its data prototypes to SwAddrMethods, and the generated Implementation Description of the RTE actually sets up this association.
symbol	Identifier	0..1	attr	Defines the section name as explained in the main description. By using this attribute for code generation (instead of the shortName) it is possible to define several different MemorySections having the same name - e.g. symbol = CODE - but using different sectionName Prefixes.

Table A.621: MemorySection

Class	MemorySectionLocation			
Package	M2::AUTOSARTemplates::CommonStructure::ResourceConsumption::ExecutionTime			
Note	Specifies in which hardware ProvidedMemorySegment the softwareMemorySection is located.			
Base	ARObject			
Aggregated by	ExecutionTime.memorySectionLocation			
Attribute	Type	Mult.	Kind	Note
provided Memory	HwElement	0..1	ref	Reference to the hardware ProvidedMemorySegment.
software MemorySection	MemorySection	0..1	ref	Reference to the MemorySection which is mapped on a certain hardware memory segment.

Table A.622: MemorySectionLocation

Class	MetaDataItem			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	This meta-class represents a single meta-data item.			
Base	ARObject			
Aggregated by	MetaDataItemSet.metaDataItem			
Attribute	Type	Mult.	Kind	Note
length	PositiveInteger	0..1	attr	This attribute determines the length of the MetaDataItem at run-time.
metaDataItem Type	TextValueSpecification	0..1	aggr	This aggregation contributes the specification of the concrete meta-data item type.

Table A.623: MetaDataItem

Class	MetaDataItemSet			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	This meta-class represents the ability to define a set of meta-data items to be used in SenderReceiver Interfaces.			
Base	ARObject			
Aggregated by	SenderReceiverInterface.metaDataItemSet			
Attribute	Type	Mult.	Kind	Note
dataElement	VariableDataPrototype	*	ref	This reference identifies the dataElement for which the ordered list of meta-data items is defined.
metaDataItem (ordered)	MetaDataItem	*	aggr	This aggregation represents the ordered definition of meta-data items.

Table A.624: MetaDataItemSet

Class	<<atpMixedString>> MixedContentForParagraph (abstract)			
Package	M2::MSR::Documentation::TextModel::InlineTextModel			
Note	This mainly represents the text model of a full blown paragraph within a documentation.			
Base	ARObject			
Subclasses	LParagraph, SlParagraph			
Attribute	Type	Mult.	Kind	Note
br	Br	1	aggr	This element is the same as function here as in a HTML document i.e. it forces a line break. **Tags:**xml.sequenceOffset=40
e	EmphasisText	1	aggr	This is emphasized text. **Tags:**xml.sequenceOffset=70
ft	SlParagraph	1	aggr	This is a foot note within a paragraph.
ie	IndexEntry	1	aggr	This is an index entry. **Tags:**xml.sequenceOffset=110
std	Std	1	aggr	This is a refeernce to a standard. **Tags:**xml.sequenceOffset=120
sub	Superscript	1	attr	This is subscript text. **Tags:**xml.sequenceOffset=100
sup	Superscript	1	attr	This is superscript text. **Tags:**xml.sequenceOffset=90
trace	Traceable	1	ref	This allows to place an arbitrary reference to a traceable object in documentation.
tt	Tt	1	aggr	This is a technical term. **Tags:**xml.sequenceOffset=30
xdoc	Xdoc	1	aggr	This is a reference to a printable external document. **Tags:**xml.sequenceOffset=130
xfile	Xfile	1	aggr	This represents a reference to an external file which usually cannot be printed. **Tags:**xml.sequenceOffset=140
xref	Xref	1	aggr	This is a cross reference. **Tags:**xml.sequenceOffset=50
xrefTarget	XrefTarget	1	aggr	This element specifies a reference target which can be scattered throughout the text. **Tags:**xml.sequenceOffset=60

Table A.625: MixedContentForParagraph

Class	ModeAccessPoint			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::ModeDeclarationGroup			
Note	A ModeAccessPoint is required by a RunnableEntity owned by a Mode Manager or Mode User. Its semantics implies the ability to access the current mode (provided by the RTE) of a ModeDeclaration GroupPrototype's ModeDeclarationGroup.			
Base	ARObject			
Aggregated by	RunnableEntity.modeAccessPoint			
Attribute	Type	Mult.	Kind	Note
ident	ModeAccessPointIdent	0..1	aggr	The aggregation in the role ident provides the ability to make the ModeAccessPoint identifiable. From the semantical point of view, the ModeAccessPoint is considered a first-class Identifiable and therefore the aggregation in the role ident shall always exist (until it may be possible to let ModeAccessPoint directly inherit from Identifiable). **Stereotypes:** atpIdentityContributor **Tags:**xml.sequenceOffset=-100
modeGroup	ModeDeclarationGroup Prototype	0..1	iref	The mode declaration group that is accessed by this runnable. **Tags:**xml.typeElement=true **InstanceRef implemented by:**ModeGroupInAtomicSwc InstanceRef

Table A.626: ModeAccessPoint

Enumeration	ModeActivationKind
Package	M2::AUTOSARTemplates::CommonStructure::ModeDeclaration
Note	Kind of mode switch condition used for activation of an event, as further described for each enumeration field.
Aggregated by	BswModeSwitchEvent.activation, SwcModeSwitchEvent.activation
Literal	Description
onEntry	On entering the referred mode. **Tags:**atp.EnumerationLiteralIndex=0
onExit	On exiting the referred mode. **Tags:**atp.EnumerationLiteralIndex=1
onTransition	On transition of the 1st referred mode to the 2nd referred mode. **Tags:**atp.EnumerationLiteralIndex=2

Table A.627: ModeActivationKind

Class	ModeDeclaration			
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, Identifiable, MultilanguageReferrable, Referrable			
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 A.628: ModeDeclaration

Class	ModeDeclarationGroup			
Package	M2::AUTOSARTemplates::CommonStructure::ModeDeclaration			
Note	A collection of Mode Declarations. Also, the initial mode is explicitly identified. **Tags:**atp.recommendedPackage=ModeDeclarationGroups			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpBlueprintable*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
initialMode	ModeDeclaration	0..1	ref	The initial mode of the ModeDeclarationGroup. This mode is active before any mode switches occurred.
mode Declaration	ModeDeclaration	*	aggr	The ModeDeclarations collected in this ModeDeclaration Group. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=modeDeclaration.shortName, mode Declaration.variationPoint.shortLabel vh.latestBindingTime=blueprintDerivationTime
modeManager ErrorBehavior	ModeErrorBehavior	0..1	aggr	This represents the ability to define the error behavior expected by the mode manager in case of errors on the mode user side (e.g. terminated mode user).
modeTransition	ModeTransition	*	aggr	This represents the avaliable ModeTransitions of the ModeDeclarationGroup
modeUserError Behavior	ModeErrorBehavior	0..1	aggr	This represents the definition of the error behavior expected by the mode user in case of errors on the mode manager side (e.g. terminated mode manager).
onTransition Value	PositiveInteger	0..1	attr	The value of this attribute shall be taken into account by the RTE generator for programmatically representing a value used for the transition between two statuses.

Table A.629: ModeDeclarationGroup

Class	ModeDeclarationGroupPrototype			
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*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, BswModuleDescription.providedModeGroup, BswModuleDescription.required ModeGroup, FirewallStateSwitchInterface.firewallStateMachine, FunctionGroupSet.functionGroup, Mode SwitchInterface.modeGroup, Process.processStateMachine, StateManagementStateNotification.state Machine			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
swCalibration Access	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 A.630: ModeDeclarationGroupPrototype

Class	ModeDeclarationGroupPrototypeMapping			
Package	M2::AUTOSARTemplates::CommonStructure::ModeDeclaration			
Note	Defines the mapping of two particular ModeDeclarationGroupPrototypes (in the given context) that are unequally named and/or require a reference to a ModeDeclarationMappingSet in order to become compatible by definition of ModeDeclarationMappings.			
Base	ARObject			
Aggregated by	ModeInterfaceMapping.modeMapping			
Attribute	Type	Mult.	Kind	Note
firstModeGroup	ModeDeclarationGroup Prototype	0..1	ref	ModeDeclarationGroupPrototype to be mapped.
mode Declaration MappingSet	ModeDeclaration MappingSet	0..1	ref	This represents the available mappings of Mode Declarations in the context ot this ModeDeclarationGroup Prototype.
secondMode Group	ModeDeclarationGroup Prototype	0..1	ref	ModeDeclarationGroupPrototype to be mapped.

Table A.631: ModeDeclarationGroupPrototypeMapping

Class	ModeDeclarationMapping			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	This meta-class implements a concrete mapping of two ModeDeclarations.			
Base	ARObject, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	AtpClassifier.atpFeature, ModeDeclarationMappingSet.modeDeclarationMapping			
Attribute	Type	Mult.	Kind	Note
firstMode	ModeDeclaration	*	ref	This represents the first ModeDeclaration of the Mode DeclarationMapping. This reference has the multiplicity 1 .. * to support use cases where e.g. one mode of the mode user is mapped to several modes of the mode manager.
secondMode	ModeDeclaration	0..1	ref	This represents the second ModeDeclaration of the Mode DeclarationMapping.

Table A.632: ModeDeclarationMapping

Class	ModeDeclarationMappingSet			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	This meta-class implements a container for ModeDeclarationGroupMappings **Tags:**atp.recommendedPackage=PortInterfaceMappingSets			
Base	ARElement, ARObject, AtpClassifier, AtpType, CollectableElement, Identifiable, Multilanguage Referrable, PackageableElement, Referrable			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
mode Declaration Mapping	ModeDeclaration Mapping	*	aggr	This represents the collection of ModeDeclaration Mappings owned by the enclosing ModeDeclaration MappingSet.

Table A.633: ModeDeclarationMappingSet

Class	ModeErrorBehavior			
Package	M2::AUTOSARTemplates::CommonStructure::ModeDeclaration			
Note	This represents the ability to define the error behavior in the context of mode handling.			
Base	ARObject			
Aggregated by	ModeDeclarationGroup.modeManagerErrorBehavior, ModeDeclarationGroup.modeUserErrorBehavior			
Attribute	Type	Mult.	Kind	Note
defaultMode	ModeDeclaration	0..1	ref	This represents the ModeDeclaration that is considered the error mode in the context of the enclosing Mode DeclarationGroup.
errorReaction Policy	ModeErrorReaction PolicyEnum	0..1	attr	This represents the ability to define the policy in terms of which default model shall apply in case an error occurs.

Table A.634: ModeErrorBehavior

Enumeration	ModeErrorReactionPolicyEnum
Package	M2::AUTOSARTemplates::CommonStructure::ModeDeclaration
Note	This represents the ability to specify the reaction on a mode error.
Aggregated by	ModeErrorBehavior.errorReactionPolicy
Literal	Description
defaultMode	This represents the ability to switch to the defaultMode in case of a mode error. **Tags:**atp.EnumerationLiteralIndex=0
lastMode	This represents the ability to keep the last mode in case of a mode error. **Tags:**atp.EnumerationLiteralIndex=1

Table A.635: ModeErrorReactionPolicyEnum

Class	ModeInBswInstanceRef			
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	ARObject, ModeInSwcBswInstanceRef			
Aggregated by	TimingModeInstance.modeInstance			
Attribute	Type	Mult.	Kind	Note
contextBsw Implementation	BswImplementation	0..1	ref	Specifies the BSW implementation that manifests the context. **Tags:**xml.sequenceOffset=10
contextMode Declaration GroupPrototype	ModeDeclarationGroup Prototype	0..1	ref	Specifies the mode declaration group prototype that manifests the context. **Tags:**xml.sequenceOffset=20
targetMode Declaration	ModeDeclaration	0..1	ref	Specifies the specific mode declaration in the given context. **Tags:**xml.sequenceOffset=30

Table A.636: ModeInBswInstanceRef

Class	ModeInSwcInstanceRef			
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	*ARObject*, *AtpInstanceRef*, *ModeInSwcBswInstanceRef*			
Aggregated by	TimingModeInstance.modeInstance			
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
context Component	SwComponent Prototype	*	ref	Specifies the SW component prototype representing the context. **Tags:**xml.sequenceOffset=20
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 A.637: ModeInSwcInstanceRef

Class	ModeInterfaceMapping			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	Defines the mapping of ModeDeclarationGroupPrototypes in context of two different ModeInterfaces.			
Base	*ARObject*, *AtpBlueprint*, *AtpBlueprintable*, *Identifiable*, *MultilanguageReferrable*, *PortInterfaceMapping*, *Referrable*			
Aggregated by	PortInterfaceMappingSet.portInterfaceMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
modeMapping	ModeDeclarationGroup PrototypeMapping	0..1	aggr	Mapping of two ModeDeclarationGroupPrototypes in two different ModeInterfaces

Table A.638: ModeInterfaceMapping

Class	ModePortAnnotation			
Package	M2::AUTOSARTemplates::SWComponentTemplate::ApplicationAttributes			
Note	Annotation to a port used for calibration regarding a certain ModeDeclarationGroupPrototype.			
Base	*ARObject*, *GeneralAnnotation*			
Aggregated by	*PortPrototype*.modePortAnnotation			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
modeGroup	ModeDeclarationGroup Prototype	0..1	ref	The instance of annotated ModeDeclarationGroup Prototype.

Table A.639: ModePortAnnotation

Class	ModeRequestTypeMap			
Package	M2::AUTOSARTemplates::CommonStructure::ModeDeclaration			
Note	Specifies a mapping between a ModeDeclarationGroup and an ImplementationDataType. This ImplementationDataType shall be used to implement the ModeDeclarationGroup.			
Base	ARObject			
Aggregated by	DataTypeMappingSet.modeRequestTypeMap			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
implementation DataType	AbstractImplementation DataType	0..1	ref	This is the corresponding AbstractImplementationData Type. It shall be modeled along the idea of an "unsigned integer-like" data type.
modeGroup	ModeDeclarationGroup	0..1	ref	This is the corresponding ModeDeclarationGroup.

Table A.640: ModeRequestTypeMap

Class	ModeSwitchEventTriggeredActivity			
Package	M2::AUTOSARTemplates::SWComponentTemplate::NvBlockComponent			
Note	This meta-class defines an activity of the NvBlockSwComponentType for a specific NvBlock which is triggered by a ModeSwitchEvent.			
Base	ARObject			
Aggregated by	NvBlockDescriptor.modeSwitchEventTriggeredActivity			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
role	Identifier	0..1	attr	This attribute indicates which service of the NvM for the NvBlock shall be requested.
swcModeSwitch Event	SwcModeSwitchEvent	0..1	ref	This reference identifies the SwcModeSwitchEvent that triggers the activity.

Table A.641: ModeSwitchEventTriggeredActivity

Class	ModeSwitchInterface			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	A mode switch interface declares a ModeDeclarationGroupPrototype to be sent and received. **Tags:**atp.recommendedPackage=PortInterfaces			
Base	*ARElement*, ARObject, *AtpBlueprint*, AtpBlueprintable, *AtpClassifier*, *AtpType*, CollectableElement, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *PortInterface*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
modeGroup	ModeDeclarationGroup Prototype	0..1	aggr	The ModeDeclarationGroupPrototype of this mode interface.

Table A.642: ModeSwitchInterface

Class	ModeSwitchPoint			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::ModeDeclarationGroup			
Note	A ModeSwitchPoint is required by a RunnableEntity owned a Mode Manager. Its semantics implies the ability to initiate a mode switch.			
Base	ARObject, *AbstractAccessPoint*, *AtpClassifier*, AtpFeature, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, RunnableEntity.modeSwitchPoint			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	ModeSwitchPoint			
modeGroup	ModeDeclarationGroup Prototype	0..1	iref	The mode declaration group that is switched by this runnable.
				InstanceRef implemented by:PModeGroupInAtomic SwcInstanceRef

Table A.643: ModeSwitchPoint

Class	ModeSwitchReceiverComSpec			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	Communication attributes of RPortPrototypes with respect to mode communication			
Base	ARObject, *RPortComSpec*			
Aggregated by	*AbstractRequiredPortPrototype*.requiredComSpec, PortPrototypeBlueprint.requiredComSpec			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
enhancedMode Api	Boolean	0..1	attr	This controls the creation of the enhanced mode API that returns information about the previous mode and the next mode. If set to "true" the enhanced mode API is supposed to be generated. For more details please refer to the SWS_RTE.
modeGroup	ModeDeclarationGroup Prototype	0..1	ref	ModeDeclarationGroupPrototype (of the same Port Interface) to which these communication attributes apply.
supports Asynchronous ModeSwitch	Boolean	0..1	attr	This attribute controls the behavior of the corresponding RPortPrototype with respect to the question whether it can deal with asynchronous mode switch requests, i.e. if set to true, the RPortPrototype is able to deal with an asynchronous mode switch request.

Table A.644: ModeSwitchReceiverComSpec

Class	ModeSwitchSenderComSpec			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	Communication attributes of PPortPrototypes with respect to mode communication			
Base	ARObject, *PPortComSpec*			
Aggregated by	*AbstractProvidedPortPrototype*.providedComSpec, PortPrototypeBlueprint.providedComSpec			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
enhancedMode Api	Boolean	0..1	attr	This controls the creation of the enhanced mode API that returns information about the previous mode and the next mode. If set to "true" the enhanced mode API is supposed to be generated. For more details please refer to the SWS_RTE.
modeGroup	ModeDeclarationGroup Prototype	0..1	ref	ModeDeclarationGroupPrototype (of the same Port Interface) to which these communication attributes apply.
modeSwitched Ack	ModeSwitchedAck Request	0..1	aggr	If this aggregation exists an acknowledgement for the successful processing of the mode switch request is required.
queueLength	PositiveInteger	0..1	attr	Length of call queue on the mode user side. The queue is implemented by the RTE. The value shall be greater or equal to 1. Setting the value of queueLength to 1 implies that incoming requests are rejected while another request that arrived earlier is being processed.

Table A.645: ModeSwitchSenderComSpec

Class	ModeSwitchedAckEvent			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents			
Note	This event is raised when the referenced ModeSwitchPoint has been processed or an error occurred.			
Base	*ARObject*, *AbstractEvent*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *Multilanguage Referrable*, *RTEEvent*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, SwcInternalBehavior.event			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
eventSource	ModeSwitchPoint	0..1	ref	The referenced ModeSwitchPoint raises this Mode SwitchedAckEvent when the ModeSwitchPoint has been processed.

Table A.646: ModeSwitchedAckEvent

Class	ModeSwitchedAckRequest			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	Requests acknowledgements that a mode switch has been proceeded successfully			
Base	*ARObject*			
Aggregated by	ModeSwitchSenderComSpec.modeSwitchedAck			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
timeout	TimeValue	0..1	attr	Number of seconds before an error is reported or in case of allowed redundancy, the value is sent again.

Table A.647: ModeSwitchedAckRequest

Class	ModeTransition			
Package	M2::AUTOSARTemplates::CommonStructure::ModeDeclaration			
Note	This meta-class represents the ability to describe possible ModeTransitions in the context of a Mode DeclarationGroup.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, ModeDeclarationGroup.modeTransition			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
enteredMode	ModeDeclaration	0..1	ref	This represents the entered model of the ModeTransition.
exitedMode	ModeDeclaration	0..1	ref	This represents the exited mode of the ModeTransition

Table A.648: ModeTransition

Class	MultidimensionalTime	
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::MultidimensionalTime	
Note	This is used to specify a multidimensional time value based on ASAM CSE codes. It is specified by a code which defined the basis of the time and a scaling factor which finally determines the time value.	
	If for example the cseCode is 100 and the cseCodeFactor is 360, it represents 360 angular degrees. If the cseCode is 0 and the cseCodeFactor is 50 it represents 50 microseconds.	
Base	*ARObject*	

▽

△

Class	MultidimensionalTime			
Aggregated by	AgeConstraint.maximum, AgeConstraint.minimum, AnalyzedExecutionTime.bestCaseExecutionTime, AnalyzedExecutionTime.worstCaseExecutionTime, ArbitraryEventTriggering.maximumDistance, ArbitraryEventTriggering.minimumDistance, BurstPatternEventTriggering.minimumInterArrivalTime, BurstPatternEventTriggering.patternJitter, BurstPatternEventTriggering.patternLength, BurstPatternEventTriggering.patternPeriod, ConcretePatternEventTriggering.offset, ConcretePatternEventTriggering.patternJitter, ConcretePatternEventTriggering.patternLength, ConcretePatternEventTriggering.patternPeriod, ConfidenceInterval.lowerBound, ConfidenceInterval.upperBound, ExecutionTimeConstraint.maximum, ExecutionTimeConstraint.minimum, IoHwAbstractionServerAnnotation.age, LatencyTimingConstraint.maximum, LatencyTimingConstraint.minimum, LatencyTimingConstraint.nominal, MeasuredExecutionTime.maximumExecutionTime, MeasuredExecutionTime.minimumExecutionTime, MeasuredExecutionTime.nominalExecutionTime, OffsetTimingConstraint.maximum, OffsetTimingConstraint.minimum, PeriodicEventTriggering.jitter, PeriodicEventTriggering.minimumInterArrivalTime, PeriodicEventTriggering.period, ReceiverAnnotation.signalAge, RoughEstimateOfExecutionTime.estimatedExecutionTime, SimulatedExecutionTime.maximumExecutionTime, SimulatedExecutionTime.minimumExecutionTime, SimulatedExecutionTime.nominalExecutionTime, SporadicEventTriggering.jitter, SporadicEventTriggering.maximumInterArrivalTime, SporadicEventTriggering.minimumInterArrivalTime, SporadicEventTriggering.period, SwDataDefProps.swRefreshTiming, SynchronizationTimingConstraint.tolerance, Trigger.triggerPeriod			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
cseCode	CseCodeType	0..1	attr	Specifies the time base by means of CSE codes.
cseCodeFactor	Integer	0..1	attr	The scaling factor for the time value based on the specified CSE code.

Table A.649: MultidimensionalTime

Class	MultilanguageReferrable (abstract)			
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::Identifiable			
Note	Instances of this class can be referred to by their identifier (while adhering to namespace borders). They also may have a longName. But they are not considered to contribute substantially to the overall structure of an AUTOSAR description. In particular it does not contain other Referrables.			
Base	ARObject, Referrable			
Subclasses	Caption, DefItem, DocumentationContext, Identifiable, SdgCaption, TraceReferrable, Traceable			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
longName	MultilanguageLong Name	0..1	aggr	This specifies the long name of the object. Long name is targeted to human readers and acts like a headline.

Table A.650: MultilanguageReferrable

Class	MultiplexedIPdu			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	A MultiplexedPdu (i.e. NOT a COM I-PDU) contains a DynamicPart, an optional StaticPart and a selector Field. In case of multiplexing this IPdu is routed between the Pdu Multiplexer and the Interface Layer.			
	A multiplexer is used to define variable parts within an IPdu that may carry different signals. The receivers of such a IPdu can determine which signalPdus are transmitted by evaluating the selector field, which carries a unique selector code for each sub-part.			
	Tags:atp.recommendedPackage=Pdus			
Base	ARObject, CollectableElement, FibexElement, IPdu, Identifiable, MultilanguageReferrable, PackageableElement, Pdu, Referrable			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*

▽

△

Class	**MultiplexedIPdu**			
dynamicPart	DynamicPart	0..1	aggr	According to the value of the selector field some parts of the IPdu have a different layout. In a complete System Description a MultiplexedIPdu shall contain a Dynamic Part. The following use cases support the multiplicity to be 0..1:
				- If a MultiplexedIPdu is received by a Pdu Gateway and is not delivered to the IPduM but routed directly to a bus interface then the content of the MulitplexedIPdu doesn't need to be described in the System Extract/Ecu Extract. - If a MultiplexedIPdu is received by an ECU which is only interested in the static part of the MultiplexedIPdu then the dynamicPart does not need to be described in the System Extract/Ecu Extract.
				atpVariation: Content of a multiplexed PDU can vary.
				Stereotypes: atpSplitable; atpVariation **Tags:** atp.Splitkey=dynamicPart, dynamicPart.variation Point.shortLabel vh.latestBindingTime=postBuild
selectorField ByteOrder	ByteOrderEnum	0..1	attr	This attribute defines the order of the bytes of the selector Field and the packing into the MultiplexedIPdu. Please consider that [constr_3247] and [constr_3223] are restricting the usage of this attribute.
				In a complete System Description this attribute is mandatory. If a MultiplexedPdu is received by a Pdu Gateway and is not delivered to the IPduM but routed directly to a bus interface then the content of the MulitplexedPdu doesn't need to be described in the System Extract/Ecu Extract. To support this use case the multiplicity is set to 0..1.
selectorField Length	Integer	0..1	attr	The size in bits of the selector field shall be configurable in a range of 1-16 bits. In a complete System Description this attribute is mandatory. If a MultiplexedPdu is received by a Pdu Gateway and is not delivered to the IPduM but routed directly to a bus interface then the content of the MulitplexedPdu doesn't need to be described in the System Extract/Ecu Extract. To support this use case the multiplicity is set to 0..1.
selectorField StartPosition	Integer	0..1	attr	This parameter is necessary to describe the position of the selector field within the IPdu.
				Note that the absolute position of the selectorField in the MultiplexedIPdu is determined by the definition of the selectorFieldByteOrder attribute of the Multiplexed Pdu. If Big Endian is specified, the start position indicates the bit position of the most significant bit in the IPdu. If Little Endian is specified, the start position indicates the bit position of the least significant bit in the IPdu. In AUTOSAR the bit counting is always set to "sawtooth" and the bit order is set to "Decreasing". The bit counting in byte 0 starts with bit 0 (least significant bit). The most significant bit in byte 0 is bit 7.
				In a complete System Description this attribute is mandatory. If a MultiplexedPdu is received by a Pdu Gateway and is not delivered to the IPduM but routed

▽

△

Class	MultiplexedIPdu				
					△ directly to a bus interface then the content of the MulitplexedPdu doesn't need to be described in the System Extract/Ecu Extract. To support this use case the multiplicity is set to 0..1.
staticPart	StaticPart	0..1	aggr		The static part of the multiplexed IPdu is the same regardless of the selector field. The static part is optional. atpVariation: Content of a multiplexed PDU can vary. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=staticPart, staticPart.variationPoint.short Label vh.latestBindingTime=postBuild
triggerMode	TriggerMode	0..1	attr		IPduM can be configured to send a transmission request for the new multiplexed IPdu to the PDU-Router because of the trigger conditions/ modes that are described in the TriggerMode enumeration. In a complete System Description this attribute is mandatory. If a MultiplexedPdu is received by a Pdu Gateway and is not delivered to the IPduM but routed directly to a bus interface then the content of the MulitplexedPdu doesn't need to be described in the System Extract/Ecu Extract. To support this use case the multiplicity is set to 0..1.
unusedBit Pattern	Integer	0..1	attr		AUTOSAR COM and AUTOSAR IPDUM are filling not used areas of an IPdu with this bit-pattern. This attribute is mandatory to avoid undefined behavior. This byte-pattern will be repeated throughout the IPdu. In a complete System Description this attribute is mandatory. If a MultiplexedPdu is received by a Pdu Gateway and is not delivered to the IPduM but routed directly to a bus interface then the content of the MulitplexedPdu doesn't need to be described in the System Extract/Ecu Extract. To support this use case the multiplicity is set to 0..1.

Table A.651: MultiplexedIPdu

Class	NPdu			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	This is a Pdu of the Transport Layer. The main purpose of the TP Layer is to segment and reassemble IPdus. **Tags:**atp.recommendedPackage=Pdus			
Base	ARObject, CollectableElement, FibexElement, IPdu, Identifiable, MultilanguageReferrable, Packageable Element, Pdu, Referrable			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.652: NPdu

Class	NetworkEndpoint				
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology				
Note	The network endpoint defines the network addressing (e.g. IP-Address or MAC multicast address).				
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*				
Aggregated by	EthernetPhysicalChannel.networkEndpoint				
Attribute	Type	Mult.	Kind	Note	
fullyQualified DomainName	String	0..1	attr	Defines the fully qualified domain name (FQDN) e.g. some.example.host.	
infrastructure Services	InfrastructureServices	0..1	aggr	Defines the network infrastructure services provided or consumed.	
ipSecConfig	IPSecConfig	0..1	aggr	Optional IPSec configuration that provides security services for IP packets.	
network Endpoint Address	NetworkEndpoint Address	1..*	aggr	Definition of a Network Address. **Tags:**xml.name Plural=NETWORK-ENDPOINT-ADDRESSES	
priority	PositiveInteger	0..1	attr	Defines the frame priority where values from 0 (best effort) to 7 (highest) are allowed.	

Table A.653: NetworkEndpoint

Class	NetworkSegmentIdentification				
Package	M2::AUTOSARTemplates::SystemTemplate::GlobalTime				
Note	This meta-class represents the ability to identify the PhysicalChannel on a system scope in a numerical way. One possible application of this approach is the Time Validation.				
Base	*ARObject*				
Aggregated by	GlobalTimeDomain.networkSegmentId				
Attribute	Type	Mult.	Kind	Note	
network SegmentId	PositiveInteger	0..1	attr	This attribute represents the numerical identifier of a PhysicalChannel on system level scope.	

Table A.654: NetworkSegmentIdentification

Class	*NmCluster* (abstract)				
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement				
Note	Set of NM nodes coordinated with use of the NM algorithm.				
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*				
Subclasses	CanNmCluster, FlexrayNmCluster, J1939NmCluster, UdpNmCluster				
Aggregated by	NmConfig.nmCluster				
Attribute	Type	Mult.	Kind	Note	
communication Cluster	CommunicationCluster	0..1	ref	Association to a CommunicationCluster in the topology description.	
nmChannel SleepMaster	Boolean	0..1	attr	This parameter shall be set to indicate if the sleep of this network can be absolutely decided by the local node only and that no other nodes can oppose that decision.	

$\bigtriangledown$

△

Class	NmCluster (abstract)			
nmNode	NmNode	*	aggr	Collection of NmNodes of the NmCluster.
				atpVariation: Derived, because NmNode can be variable.
				Stereotypes: atpSplitable; atpVariation **Tags:** atp.Splitkey=nmNode.shortName, nmNode.variation Point.shortLabel vh.latestBindingTime=postBuild
nmNode Detection Enabled	Boolean	0..1	attr	Enables the Request Repeat Message Request support. Only valid if nmNodeIdEnabled is set to true.
nmNodeId Enabled	Boolean	0..1	attr	Enables the source node identifier.
nmPnc Participation	Boolean	0..1	attr	Defines whether this NmCluster contributes to the partial network mechanism.
nmRepeatMsg IndEnabled	Boolean	0..1	attr	Switch for enabling the Repeat Message Bit Indication.
nm Synchronizing Network	Boolean	0..1	attr	If this parameter is true, then this network is a synchronizing network for the NM coordination cluster which it belongs to. The network is expected to call Nm_ SynchronizationPoint() at regular intervals.
pncCluster VectorLength	PositiveInteger	0..1	attr	Optionally defines the length of the PNC Vector per CommunicationCluster (and VLAN in case of UdpNm). If not defined then System.pncVectorLength applies.
				Should only make the PNC Vector shorter (or same length as defined in System.pncVectorLength).
				Tags:atp.Status=draft

Table A.655: NmCluster

Class	NmConfig			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	Contains the all configuration elements for AUTOSAR Nm.			
	Tags:atp.recommendedPackage=NmConfigs			
Base	ARObject, CollectableElement, FibexElement, Identifiable, MultilanguageReferrable, Packageable Element, Referrable			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
nmCluster	NmCluster	*	aggr	Collection of NM Clusters
				atpVariation: Derived, because cluster can be variable.
				Stereotypes: atpSplitable; atpVariation **Tags:** atp.Splitkey=nmCluster.shortName, nmCluster.variation Point.shortLabel vh.latestBindingTime=postBuild
nmCluster Coupling	NmClusterCoupling	*	aggr	Collection of NmClusterCouplings
				atpVariation: Derived, because NmCluster can vary.
				Stereotypes: atpSplitable; atpVariation **Tags:** atp.Splitkey=nmClusterCoupling, nmCluster Coupling.variationPoint.shortLabel vh.latestBindingTime=postBuild

▽

△

Class	NmConfig			
nmIfEcu	NmEcu	*	aggr	Collection of NM ECUs
				atpVariation: Derived, because EcuInstance can be variable.
				Stereotypes: atpSplitable; atpVariation **Tags:** atp.Splitkey=nmIfEcu.shortName, nmIfEcu.variation Point.shortLabel vh.latestBindingTime=preCompileTime

Table A.656: NmConfig

Class	**NmEcu**			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	ECU on which NM is running.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	NmConfig.nmIfEcu			
Attribute	Type	Mult.	Kind	Note
busDependent NmEcu	BusspecificNmEcu	*	aggr	Cluster specific NmEcu attributes
ecuInstance	EcuInstance	1	ref	Association to an ECUInstance in the topology description.
nmBus Synchronization Enabled	Boolean	0..1	attr	Enables bus synchronization support.
nmComControl Enabled	Boolean	0..1	attr	Enables the Communication Control support.
nmCoordinator	NmCoordinator	0..1	aggr	Nm ECU may coordinate different clusters.
nmCycletime MainFunction	TimeValue	0..1	attr	The period between successive calls to the Main Function of the NM Interface in seconds.
nmPduRx Indication Enabled	Boolean	0..1	attr	Switch for enabling the PDU Rx Indication.
nmRemote SleepInd Enabled	Boolean	0..1	attr	Switch for enabling remote sleep indication support.
nmStateChange IndEnabled	Boolean	0..1	attr	Enables the CAN Network Management state change notification.
nmUserData Enabled	Boolean	0..1	attr	Switch for enabling user data support.

Table A.657: NmEcu

Class	**NmNode** (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	The linking of NmEcus to NmClusters is realized via the NmNodes.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	CanNmNode, FlexrayNmNode, J1939NmNode, UdpNmNode			
Aggregated by	NmCluster.nmNode			
Attribute	Type	Mult.	Kind	Note
controller	Communication Controller	0..1	ref	Association to an CommunicationController in the topology description.

▽

△

Class	NmNode (abstract)			
nmCoord Cluster	PositiveInteger	0..1	attr	NmCoordinationCluster identification number.
nmCoordinator Role	NmCoordinatorRole Enum	0..1	attr	This attribute indicates the role the NM Coordinator will have on this channel.
nmIfEcu	NmEcu	0..1	ref	Reference to the NmEcu that contains this NmNode. (CommunicationController that is referenced by the Nm Node shall be contained in the EcuInstance that is referenced by the NmEcu).
nmNodeId	Integer	0..1	attr	Node identifier of local NmNode. Shall be unique in the NmCluster.
nmPassive ModeEnabled	Boolean	0..1	attr	Enables support of the Passive Mode. The passive mode is configurable per channel.
rxNmPdu	NmPdu	*	ref	receive NM Pdu.
txNmPdu	NmPdu	*	ref	transmit NM Pdu

Table A.658: NmNode

Class	NmPdu			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Network Management Pdu **Tags:**atp.recommendedPackage=Pdus			
Base	ARObject, CollectableElement, FibexElement, Identifiable, MultilanguageReferrable, Packageable Element, Pdu, Referrable			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
iSignalToIPdu Mapping	ISignalToIPduMapping	*	aggr	This optional aggregation is used to describe NmUser Data that is transmitted in the NmPdu. The counting of the startPosition starts at the beginning of the NmPdu regardless whether Cbv or Nid are used.
nmData Information	Boolean	0..1	attr	Defines if the Pdu contains NM Data. If the NmPdu does not aggregate any ISignalToIPduMappings it still may contain UserData that is set via Nm_SetUserData(). If the ISignalToIPduMapping exists then the nmDataInformation attribute shall be ignored.
nmVote Information	Boolean	0..1	attr	Defines if the Pdu contains NM Vote information.
unusedBit Pattern	Integer	0..1	attr	AUTOSAR COM is filling not used areas of an Pdu with this bit-pattern. This attribute can only be used if the nm DataInformation attribute is set to true.

Table A.659: NmPdu

Class	NonqueuedReceiverComSpec			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	Communication attributes specific to non-queued receiving.			
Base	ARObject, RPortComSpec, ReceiverComSpec			
Aggregated by	AbstractRequiredPortPrototype.requiredComSpec, PortPrototypeBlueprint.requiredComSpec			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	NonqueuedReceiverComSpec			
aliveTimeout	TimeValue	0..1	attr	Specify the amount of time (in seconds) after which the software component (via the RTE) needs to be notified if the corresponding data item have not been received according to the specified timing description. If the aliveTimeout attribute is 0 no timeout monitoring shall be performed.
enableUpdate	Boolean	0..1	attr	This attribute controls whether application code is entitled to check whether the value of the corresponding Variable DataPrototype has been updated.
filter	DataFilter	0..1	aggr	The applicable filter algorithm for filtering the value of the corresponding dataElement.
handleData Status	Boolean	0..1	attr	If this attribute is set to true, then the Rte_IStatus API shall exist. If the attribute does not exist or is set to false, then the Rte_IStatus API may still exist in response to the existence of further conditions.
handleNever Received	Boolean	0..1	attr	This attribute specifies whether for the corresponding VariableDataPrototype the "never received" flag is available. If yes, the RTE is supposed to assume that initially the VariableDataPrototype has not been received before. After the first reception of the corresponding VariableDataPrototype the flag is cleared. • If the value of this attribute is set to "true" the flag is required. • If set to "false", the RTE shall not support the "never received" functionality for the corresponding VariableDataPrototype.
handleTimeout Type	HandleTimeoutEnum	0..1	attr	This attribute controls the behavior with respect to the handling of timeouts.
initValue	ValueSpecification	0..1	aggr	Initial value to be used in case the sending component is not yet initialized. If the sender also specifies an initial value, then the receiver's value will be used.
timeout Substitution Value	ValueSpecification	0..1	aggr	This attribute represents the substitution value applicable in the case of a timeout.

Table A.660: NonqueuedReceiverComSpec

Class	NonqueuedSenderComSpec			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	Communication attributes for non-queued sender/receiver communication (sender side)			
Base	ARObject, PPortComSpec, SenderComSpec			
Aggregated by	AbstractProvidedPortPrototype.providedComSpec, PortPrototypeBlueprint.providedComSpec			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dataFilter	DataFilter	0..1	aggr	The applicable filter algorithm for filtering the value of the corresponding dataElement.
initValue	ValueSpecification	0..1	aggr	Initial value to be sent if sender component is not yet fully initialized, but receiver needs data already.

Table A.661: NonqueuedSenderComSpec

Class	NotAvailableValueSpecification
Package	M2::AUTOSARTemplates::CommonStructure::Constants
Note	This meta-class provides the ability to specify a ValueSpecification to state that the respective element is not available. This ability is needed to support the existence of ApplicationRecordElements where attribute isOptional ist set to the value true. **Tags:**atp.Status=draft
Base	ARObject, *ValueSpecification*
Aggregated by	ApplicationAssocMapElementValueSpecification.key, ApplicationAssocMapElementValueSpecification. value, ArrayValueSpecification.element, CalibrationParameterValue.applInitValue, CalibrationParameter Value.implInitValue, ConstantSpecification.valueSpec, CryptoServiceKey.developmentValue, Diagnostic EnvDataCondition.compareValue, DiagnosticEnvDataElementCondition.compareValue, FieldSenderCom Spec.initValue, ISignal.initValue, ISignal.timeoutSubstitutionValue, NonqueuedReceiverComSpec.init Value, NonqueuedReceiverComSpec.timeoutSubstitutionValue, NonqueuedSenderComSpec.initValue, NvProvideComSpec.ramBlockInitValue, NvProvideComSpec.romBlockInitValue, NvRequireComSpec.init Value, ParameterDataPrototype.initValue, ParameterProvideComSpec.initValue, ParameterRequireCom Spec.initValue, PersistencyDataRequiredComSpec.initValue, PersistencyKeyValuePair.initValue, Port DefinedArgumentValue.value, PortPrototypeBlueprintInitValue.value, RecordValueSpecification.field, *StateManagementCompareCondition*.compareValue, SwDataDefProps.invalidValue, VariableData Prototype.initValue

Attribute	Type	Mult.	Kind	Note
defaultPattern	PositiveInteger	0..1	attr	The content of this attribute shall be used to initialize gaps in the memory occupied by a structured data type in the case that an NotAvailableValueSpecification is used. Note that this pattern is only applied during initialization!

Table A.662: NotAvailableValueSpecification

Primitive	Numerical
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::PrimitiveTypes
Note	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. The value can be expressed in octal, hexadecimal, binary representation. Negative numbers can only be expressed in decimal or float notation. **Tags:** 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]+)?)([eE]([+\-]?)[0-9]+)?)|\.0|INF|-INF|NaN xml.xsd.type=string

Table A.663: Numerical

Class	NumericalOrText
Package	M2::AUTOSARTemplates::CommonStructure::Constants
Note	This meta-class represents the ability to yield either a numerical or a string. A typical use case is that two or more instances of this meta-class are aggregated with a VariationPoint where some instances yield strings while other instances yield numerical depending on the resolution of the binding expression.
Base	ARObject
Aggregated by	RuleArguments.vtf, SwValues.vtf

Attribute	Type	Mult.	Kind	Note

▽

△

Class	NumericalOrText			
vf	Numerical	0..1	attr	This attribute represents the ability to provide a numerical value. The latest binding time of the VariationPoint shall be preCompileTime. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=10
vt	String	0..1	attr	This attribute represents the ability to provide a textual value. **Tags:**xml.sequenceOffset=20

Table A.664: NumericalOrText

Class	NumericalRuleBasedValueSpecification			
Package	M2::AUTOSARTemplates::CommonStructure::Constants			
Note	This meta-class is used to support a rule-based initialization approach for data types with an array-nature (ImplementationDataType of category ARRAY).			
Base	*ARObject*, *AbstractRuleBasedValueSpecification*, *ValueSpecification*			
Aggregated by	ApplicationAssocMapElementValueSpecification.key, ApplicationAssocMapElementValueSpecification. value, ArrayValueSpecification.element, CalibrationParameterValue.applInitValue, CalibrationParameter Value.implInitValue, ConstantSpecification.valueSpec, CryptoServiceKey.developmentValue, Diagnostic EnvDataCondition.compareValue, DiagnosticEnvDataElementCondition.compareValue, FieldSenderCom Spec.initValue, ISignal.initValue, ISignal.timeoutSubstitutionValue, NonqueuedReceiverComSpec.init Value, NonqueuedReceiverComSpec.timeoutSubstitutionValue, NonqueuedSenderComSpec.initValue, NvProvideComSpec.ramBlockInitValue, NvProvideComSpec.romBlockInitValue, NvRequireComSpec.init Value, ParameterDataPrototype.initValue, ParameterProvideComSpec.initValue, ParameterRequireCom Spec.initValue, PersistencyDataRequiredComSpec.initValue, PersistencyKeyValuePair.initValue, Port DefinedArgumentValue.value, PortPrototypeBlueprintInitValue.value, RecordValueSpecification.field, *StateManagementCompareCondition*.compareValue, SwDataDefProps.invalidValue, VariableData Prototype.initValue			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
ruleBased Values	RuleBasedValue Specification	0..1	aggr	This represents the rule based value specification for the array. **Tags:** xml.roleElement=true xml.roleWrapperElement=false xml.typeWrapperElement=false

Table A.665: NumericalRuleBasedValueSpecification

Class	NumericalValueSpecification
Package	M2::AUTOSARTemplates::CommonStructure::Constants
Note	A numerical ValueSpecification which is intended to be assigned to a Primitive data element. Note that the numerical value is a variant, it can be computed by a formula.
Base	*ARObject*, *ValueSpecification*

▽

△

Class	NumericalValueSpecification
Aggregated by	ApplicationAssocMapElementValueSpecification.key, ApplicationAssocMapElementValueSpecification. value, ArrayValueSpecification.element, CalibrationParameterValue.applInitValue, CalibrationParameter Value.implInitValue, ConstantSpecification.valueSpec, CryptoServiceKey.developmentValue, Diagnostic EnvDataCondition.compareValue, DiagnosticEnvDataElementCondition.compareValue, FieldSenderCom Spec.initValue, ISignal.initValue, ISignal.timeoutSubstitutionValue, NonqueuedReceiverComSpec.init Value, NonqueuedReceiverComSpec.timeoutSubstitutionValue, NonqueuedSenderComSpec.initValue, NvProvideComSpec.ramBlockInitValue, NvProvideComSpec.romBlockInitValue, NvRequireComSpec.init Value, ParameterDataPrototype.initValue, ParameterProvideComSpec.initValue, ParameterRequireCom Spec.initValue, PersistencyDataRequiredComSpec.initValue, PersistencyKeyValuePair.initValue, Port DefinedArgumentValue.value, PortPrototypeBlueprintInitValue.value, RecordValueSpecification.field, *StateManagementCompareCondition*.compareValue, SwDataDefProps.invalidValue, VariableData Prototype.initValue

Attribute	Type	Mult.	Kind	Note
value	Numerical	0..1	attr	This is the value itself. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime

Table A.666: NumericalValueSpecification

Class	NvBlockDataMapping
Package	M2::AUTOSARTemplates::SWComponentTemplate::NvBlockComponent
Note	Defines the mapping between the VariableDataPrototypes in the NvBlockComponents ports and the VariableDataPrototypes of the RAM Block. The data types of the referenced VariableDataPrototypes in the ports and the referenced sub-element (inside a CompositeDataType) of the VariableDataPrototype representing the RAM Block shall be compatible.
Base	*ARObject*
Aggregated by	BulkNvDataDescriptor.nvBlockDataMapping, NvBlockDescriptor.nvBlockDataMapping

Attribute	Type	Mult.	Kind	Note
bitfieldTextTable MaskNvBlock Descriptor	PositiveInteger	0..1	attr	This attribute identifies the applicable bit mask on the side of the Nv Block.
bitfieldTextTable MaskPort Prototype	PositiveInteger	0..1	attr	This attribute identifies the applicable bit mask on the side of the PortPrototype.
nvRamBlock Element	AutosarVariableRef	0..1	aggr	Reference to a VariableDataPrototype of a RAM Block.
readNvData	AutosarVariableRef	0..1	aggr	Reference to a VariableDataPrototype of a pPort of the NvBlockComponent providing read access to the RAM Block.If there is no PortPrototype providing read access (write-only) the reference can be omitted.
writtenNvData	AutosarVariableRef	0..1	aggr	Reference to a VariableDataPrototype of a rPort of the Nv BlockComponent providing write access to the RAM Block. If there is no port providing write access (read-only) the reference can be omitted.
writtenReadNv Data	AutosarVariableRef	0..1	aggr	Reference to a VariableDataPrototype of a PRPort Prototype of the NvBlockSwComponentType providing write and read access to the RAM Block.

Table A.667: NvBlockDataMapping

Class	NvBlockDescriptor			
Package	M2::AUTOSARTemplates::SWComponentTemplate::NvBlockComponent			
Note	Specifies the properties of exactly on NVRAM Block.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, NvBlockSwComponentType.nvBlockDescriptor			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
clientServerPort	RoleBasedPort Assignment	*	aggr	The RoleBasedPortAssignement defines which client server port of the NvBlockSwComponentType serves for which kind of service or notification. In case of notifications one common callback function is provided by the RTE for each individual kind of notification defined by the "role". The aggregation of RoleBasedPortAssignment is subject to variability with the purpose to support the conditional existence of ports. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=clientServerPort, clientServerPort.variation Point.shortLabel vh.latestBindingTime=preCompileTime
constantValue Mapping	ConstantSpecification MappingSet	*	ref	Reference to the ConstantSpecificationMapping to be applied for the particular NVRAM Block **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=constantValueMapping
dataType Mapping	DataTypeMappingSet	*	ref	Reference to the DataTypeMapping to be applied for the particular NVRAM Block. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=dataTypeMapping
instantiation DataDefProps	InstantiationDataDef Props	*	aggr	The purpose of InstantiationDataDefProps are the refinement of some data def properties of individual instantiations within the context of a NvBlockSw ComponentType. The aggregation of InstantiationDataDefProps is subject to variability with the purpose to support the conditional existence of ports, component internal memory objects and those attributes. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=instantiationDataDefProps, instantiationData DefProps.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
modeSwitch EventTriggered Activity	ModeSwitchEvent TriggeredActivity	*	aggr	This represents the collection of ModeSwitchEvent TriggeredActivities related to the enclosing NvBlock Descriptor. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=modeSwitchEventTriggeredActivity, mode SwitchEventTriggeredActivity.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	NvBlockDescriptor				
nvBlockData Mapping	NvBlockDataMapping	*	aggr	Defines the mapping between the VariableData Prototypes in the NvBlockComponents ports and the VariableDataPrototypes of the RAM Block. The aggregation of NvBlockDataMapping is subject to variability with the purpose to support the conditional existence of nv data ports. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=nvBlockDataMapping, nvBlockData Mapping.variationPoint.shortLabel vh.latestBindingTime=preCompileTime	
nvBlockNeeds	NvBlockNeeds	0..1	aggr	Specifies the abstract needs on the configuration of the NVRAM Manager for the single NVRAM Block described by this NvBlockDescriptor. In addition, it may define requirements for writing strategies in an implementation of an NvBlockSw ComponentType by the RTE. Please note that the attributes nDataSets and nRom Blocks are not relevant for this aggregation because the RTE will allocate just one block anyway. In a different context, however, they do make sense.	
ramBlock	VariableDataPrototype	0..1	aggr	Defines the RAM Block of the NVRAM Block provided by NvBlockSwComponentType.	
romBlock	ParameterData Prototype	0..1	aggr	Defines the ROM Block of the NVRAM Block provided by NvBlockSwComponentType.	
supportDirty Flag	Boolean	0..1	attr	Specifies whether calling of NvM functions for writing and/ or status control of potentially modified RAM Blocks to NV memory shall be controlled by the RTE.	
timingEvent	TimingEvent	0..1	ref	this reference can be taken to identify the TimingEvent to be used by the RTE for implementing a cyclic writing strategy for this block	
writingStrategy	RoleBasedData Assignment	*	aggr	This attribute allows for assigning a specific writing strategy for an incoming AutosarDataPrototype.	

Table A.668: NvBlockDescriptor

Class	NvBlockNeeds				
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds				
Note	Specifies the abstract needs on the configuration of a single NVRAM Block.				
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *ServiceNeeds*				
Aggregated by	BswServiceDependency.serviceNeeds, NvBlockDescriptor.nvBlockNeeds, SwcServiceDependency.serviceNeeds				
Attribute	*Type*	*Mult.*	*Kind*	*Note*	
calcRamBlock Crc	Boolean	0..1	attr	Defines if CRC (re)calculation for the permanent RAM Block is required.	
checkStatic BlockId	Boolean	0..1	attr	Defines if the Static Block Id check shall be enabled.	
cyclicWriting Period	TimeValue	0..1	attr	This represents the period for cyclic writing of NvData to store the associated RAM Block.	
nDataSets	PositiveInteger	0..1	attr	Number of data sets to be provided by the NVRAM manager for this block. This is the total number of ROM Blocks and RAM Blocks.	

▽

△

Class	NvBlockNeeds			
nRomBlocks	PositiveInteger	0..1	attr	Number of ROM Blocks to be provided by the NVRAM manager for this block. Please note that these multiple ROM Blocks are given in a contiguous area.
ramBlockStatus Control	RamBlockStatusControl Enum	0..1	attr	This attribute defines how the management of the RAM Block status is controlled.
readonly	Boolean	0..1	attr	true: data of this NVRAM Block are write protected for normal operation (but protection can be disabled) false: no restriction
reliability	NvBlockNeeds ReliabilityEnum	0..1	attr	Reliability against data loss on the non-volatile medium.
resistantTo ChangedSw	Boolean	0..1	attr	Defines whether an NVRAM Block shall be treated resistant to configuration changes (true) or not (false). For details how to handle initialization in the latter case, please refer to the NVRAM specification.
restoreAtStart	Boolean	0..1	attr	Defines whether the associated RAM Block shall be implicitly restored during startup by the basic software.
selectBlockFor FirstInitAll	Boolean	0..1	attr	If this attribute is set to true the NvM shall process this block in the NvM_FirstInitAll() function.
storeAt Shutdown	Boolean	0..1	attr	Defines whether or not the associated RAM Block shall be implicitly stored during shutdown by the basic software.
storeCyclic	Boolean	0..1	attr	Defines whether or not the associated RAM Block shall be implicitly stored periodically by the basic software.
store Emergency	Boolean	0..1	attr	Defines whether or not the associated RAM Block shall be implicitly stored in case of ECU failure (e.g. loss of power) by the basic software. If the attribute store Emergency is set to true the associated RAM Block shall be configured to have immediate priority.
storeImmediate	Boolean	0..1	attr	Defines whether or not the associated RAM Block shall be implicitly stored immediately during or after execution of the according SW-C RunnableEntity by the basic software.
storeOnChange	Boolean	0..1	attr	This attribute defines whether the associated RAM Block shall be stored immediately if the written value is different to the value stored in the associated RAM Block(s) during or after execution of the according SW-C RunnableEntity.
useAuto ValidationAt ShutDown	Boolean	0..1	attr	If set to true the RAM Block shall be auto validated during shutdown phase.
useCRCComp Mechanism	Boolean	0..1	attr	If set to true the CRC of the RAM Block shall be compared during a write job with the CRC which was calculated during the last successful read or write job in order to skip unnecessary NVRAM writings.
writeOnlyOnce	Boolean	0..1	attr	Defines write protection after first write: true: This block is prevented from being changed/erased or being replaced with the default ROM data after first initialization by the software-component. false: No such restriction.
writeVerification	Boolean	0..1	attr	Defines if Write Verification shall be enabled for this NVRAM Block.
writing Frequency	PositiveInteger	0..1	attr	Provides the amount of updates to this block from the application point of view. It has to be provided in "number of write access per year".
writingPriority	NvBlockNeedsWriting PriorityEnum	0..1	attr	Requires the priority of writing this block in case of concurrent requests to write other blocks.

Table A.669: NvBlockNeeds

Enumeration	NvBlockNeedsReliabilityEnum
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds
Note	Reliability against data loss on the non-volatile medium. These requirements give only a relative indication, for example on the required degree of redundancy for storage.
	They do, however, not specify by which means (e.g. software or hardware) the reliability is actually achieved.
Aggregated by	NvBlockNeeds.reliability
Literal	Description
errorCorrection	Errors shall be corrected
	Tags:atp.EnumerationLiteralIndex=0
errorDetection	Errors shall be detected
	Tags:atp.EnumerationLiteralIndex=1
noProtection	Data need not to be handled with protection
	Tags:atp.EnumerationLiteralIndex=2

Table A.670: NvBlockNeedsReliabilityEnum

Class	NvBlockSwComponentType			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components			
Note	The NvBlockSwComponentType defines non volatile data which data can be shared between Sw ComponentPrototypes. The non volatile data of the NvBlockSwComponentType are accessible via provided and required ports.			
	Tags:atp.recommendedPackage=SwComponentTypes			
Base	*ARElement*, *ARObject*, *AtomicSwComponentType*, *AtpBlueprint*, *AtpBlueprintable*, *AtpClassifier*, *Atp Type*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *Sw ComponentType*			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
bulkNvData Descriptor	BulkNvDataDescriptor	*	aggr	This aggregation formally defines the bulk Nv Blocks that are provided to the application software by the enclosing NvBlockSwComponentType.
				Stereotypes: atpSplitable; atpVariation **Tags:** atp.Splitkey=bulkNvDataDescriptor.shortName, bulkNv DataDescriptor.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
nvBlock Descriptor	NvBlockDescriptor	*	aggr	Specification of the properties of exactly one NVRAM Block.
				Stereotypes: atpSplitable; atpVariation **Tags:** atp.Splitkey=nvBlockDescriptor.shortName, nvBlock Descriptor.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table A.671: NvBlockSwComponentType

Class	NvDataInterface
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface
Note	A non volatile data interface declares a number of VariableDataPrototypes to be exchanged between non volatile block components and atomic software components.
	Tags:atp.recommendedPackage=PortInterfaces

$\bigtriangledown$

△

Class	NvDataInterface			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpBlueprintable*, *AtpClassifier*, *AtpType*, *CollectableElement*, *DataInterface*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *PortInterface*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
nvData	VariableDataPrototype	*	aggr	The VariableDataPrototype of this nv data interface.

Table A.672: NvDataInterface

Class	NvDataPortAnnotation			
Package	M2::AUTOSARTemplates::SWComponentTemplate::ApplicationAttributes			
Note	Annotation to a port regarding a certain VariableDataPrototype.			
Base	*ARObject*, *GeneralAnnotation*			
Aggregated by	*PortPrototype*.nvDataPortAnnotation			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
variable	VariableDataPrototype	0..1	ref	The instance of nv data annotated.

Table A.673: NvDataPortAnnotation

Class	NvProvideComSpec			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	Communication attributes of PPortPrototypes with respect to Nv data communication on the provided side.			
Base	*ARObject*, *PPortComSpec*			
Aggregated by	*AbstractProvidedPortPrototype*.providedComSpec, PortPrototypeBlueprint.providedComSpec			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
ramBlockInit Value	ValueSpecification	0..1	aggr	This represents the initial value of the RAM Block that corresponds to the referenced variable.
romBlockInit Value	ValueSpecification	0..1	aggr	This represents the initial value of the ROM block that corresponds to the referenced variable.
variable	VariableDataPrototype	0..1	ref	This represents the variable for which the ComSpec is specified.

Table A.674: NvProvideComSpec

Class	NvRequireComSpec			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	Communication attributes of RPortPrototypes with respect to Nv data communication on the required side.			
Base	*ARObject*, *RPortComSpec*			
Aggregated by	*AbstractRequiredPortPrototype*.requiredComSpec, PortPrototypeBlueprint.requiredComSpec			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
initValue	ValueSpecification	0..1	aggr	The initial value owned by the NvComSpec
variable	VariableDataPrototype	0..1	ref	The VariableDataPrototype the ComSpec applies for.

Table A.675: NvRequireComSpec

Class	ObdControlServiceNeeds			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	Specifies the abstract needs of a component or module on the configuration of OBD Service 08 (request control of on-board system) in relation to a particular test-Identifier (TID) supported by this component or module.			
Base	ARObject, DiagnosticCapabilityElement, Identifiable, MultilanguageReferrable, Referrable, Service Needs			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.676: ObdControlServiceNeeds

Class	ObdInfoServiceNeeds			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	Specifies the abstract needs of a component or module on the configuration of OBD Services in relation to a given InfoType (OBD Service 09) which is supported by this component or module.			
Base	ARObject, DiagnosticCapabilityElement, Identifiable, MultilanguageReferrable, Referrable, Service Needs			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.677: ObdInfoServiceNeeds

Class	ObdPidServiceNeeds			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	Specifies the abstract needs of a component or module on the configuration of OBD Services in relation to a particular PID (parameter identifier) which is supported by this component or module. In case of using a client/server communicated value, the related value shall be communicated via the port referenced by assignedPort. The details of this communication (e.g. appropriate naming conventions) are specified in the related software specifications (SWS).			
Base	ARObject, DiagnosticCapabilityElement, Identifiable, MultilanguageReferrable, Referrable, Service Needs			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.678: ObdPidServiceNeeds

Enumeration	ObdRatioConnectionKindEnum
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds
Note	Defines the way how the IUMPR service connection between the Dem and the client component or module is handled (for details see the DEM Specification).
Aggregated by	ObdRatioServiceNeeds.connectionType
Literal	Description
apiUse	The IUMPR service (of the DEM) uses an explicit API to connect to the component or module. **Tags:**atp.EnumerationLiteralIndex=0
observer	The IUMPR service (of the Dem) uses no API but "observes" the associated diagnostic event. **Tags:**atp.EnumerationLiteralIndex=1

Table A.679: ObdRatioConnectionKindEnum

Class	ObdRatioServiceNeeds			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	Specifies the abstract needs of a component or module on the configuration of OBD Services in relation to a particular "ratio monitoring" which is supported by this component or module.			
Base	*ARObject*, *DiagnosticCapabilityElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *Service Needs*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
connectionType	ObdRatioConnection KindEnum	0..1	attr	Defines how the DEM is connected to the component or module to perform the IUMPR (In use monitor performance ratio) service.
rateBased MonitoredEvent	DiagnosticEventNeeds	0..1	ref	The rate based monitored Diagnostic Event.
usedFid	FunctionInhibitionNeeds	0..1	ref	This represents the primary Function Inhibition Identifier used for the rate based monitor. This is an optional attribute.

Table A.680: ObdRatioServiceNeeds

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 mimum 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 A.681: OffsetTimingConstraint

Class	OperationInSystemInstanceRef			
Package	M2::AUTOSARTemplates::SystemTemplate::InstanceRefs			
Note				
Base	*ARObject*, *AtpInstanceRef*			
Aggregated by	ClientIdDefinition.clientServerOperation, ClientServerToSignalMapping.clientServerOperation, PortElementToCommunicationResourceMapping.clientServerOperation, SwcToSwcOperationArguments.operation			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	OperationInSystemInstanceRef			
base	System	0..1	ref	**Stereotypes:** atpDerived **Tags:**xml.sequenceOffset=10
context Component (ordered)	SwComponent Prototype	*	ref	**Tags:**xml.sequenceOffset=30
context Composition	RootSwComposition Prototype	0..1	ref	**Tags:**xml.sequenceOffset=20
contextPort	PortPrototype	1	ref	**Tags:**xml.sequenceOffset=40
targetOperation	ClientServerOperation	1	ref	**Tags:**xml.sequenceOffset=50

Table A.682: OperationInSystemInstanceRef

Class	OperationInvokedEvent			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents			
Note	This event is raised when the ClientServerOperation referenced in OperationInvokedEvent.operation shall be invoked.			
Base	*ARObject*, *AbstractEvent*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *Multilanguage Referrable*, *RTEEvent*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, SwcInternalBehavior.event			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
operation	ClientServerOperation	0..1	iref	This represents the ClientServerOperation which shall be invoked. **InstanceRef implemented by:**POperationInAtomicSwc InstanceRef

Table A.683: OperationInvokedEvent

Class	OsTaskExecutionEvent			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents			
Note	This RTEEvent is supposed to execute RunnableEntities which have to react on the execution of specific OsTasks. Therefore, this event is unconditionally raised whenever the OsTask on which it is mapped is executed. The main use case for this event is scheduling of Runnables of Complex Drivers which have to react on task executions. **Tags:**atp.Status=draft			
Base	*ARObject*, *AbstractEvent*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *Multilanguage Referrable*, *RTEEvent*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, SwcInternalBehavior.event			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.684: OsTaskExecutionEvent

Class	PPortComSpec (abstract)	
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication	
Note	Communication attributes of a provided PortPrototype. This class will contain attributes that are valid for all kinds of provide ports, independent of client-server or sender-receiver communication patterns.	
Base	*ARObject*	

▽

△

Class	PPortComSpec (abstract)			
Subclasses	ModeSwitchSenderComSpec, NvProvideComSpec, ParameterProvideComSpec, SenderComSpec, ServerComSpec			
Aggregated by	AbstractProvidedPortPrototype.providedComSpec, PortPrototypeBlueprint.providedComSpec			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.685: PPortComSpec

Class	PPortPrototype			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components			
Note	Component port providing a certain port interface.			
Base	ARObject, AbstractProvidedPortPrototype, AtpBlueprintable, AtpFeature, AtpPrototype, Identifiable, MultilanguageReferrable, PortPrototype, Referrable			
Aggregated by	AtpClassifier.atpFeature, SwComponentType.port			
Attribute	Type	Mult.	Kind	Note
provided Interface	PortInterface	0..1	tref	The interface that this port provides. **Stereotypes:** isOfType

Table A.686: PPortPrototype

Class	PRPortPrototype			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components			
Note	This kind of PortPrototype can take the role of both a required and a provided PortPrototype.			
Base	ARObject, AbstractProvidedPortPrototype, AbstractRequiredPortPrototype, AtpBlueprintable, AtpFeature, AtpPrototype, Identifiable, MultilanguageReferrable, PortPrototype, Referrable			
Aggregated by	AtpClassifier.atpFeature, SwComponentType.port			
Attribute	Type	Mult.	Kind	Note
provided Required Interface	PortInterface	0..1	tref	This represents the PortInterface used to type the PRPort Prototype **Stereotypes:** isOfType

Table A.687: PRPortPrototype

Class	PackageableElement (abstract)			
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::ARPackage			
Note	This meta-class specifies the ability to be a member of an AUTOSAR package.			
Base	ARObject, CollectableElement, Identifiable, MultilanguageReferrable, Referrable			
Subclasses	ARElement, EnumerationMappingTable, FibexElement			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.688: PackageableElement

Class	ParameterAccess			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::DataElements			
Note	The presence of a ParameterAccess implies that a RunnableEntity needs access to a ParameterData Prototype.			
Base	*ARObject*, *AbstractAccessPoint*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, RunnableEntity.parameterAccess			
Attribute	Type	Mult.	Kind	Note
accessed Parameter	AutosarParameterRef	0..1	aggr	Reference to the accessed calibration parameter.
swDataDef Props	SwDataDefProps	0..1	aggr	This allows denote instance and access specific properties, mainly input values and common axis. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=swDataDefProps

Table A.689: ParameterAccess

Class	ParameterDataPrototype			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::DataPrototypes			
Note	A ParameterDataPrototype represents a formalized generic piece of information that is typically immutable by the application software layer, but mutable by measurement and calibration tools. ParameterDataPrototype is used in various contexts and the specific context gives the otherwise generic ParameterDataPrototype a dedicated semantics.			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *AutosarDataPrototype*, *DataPrototype*, *Identifiable*, *Multilanguage Referrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, BswInternalBehavior.perInstanceParameter, *InternalBehavior*.constant Memory, NvBlockDescriptor.romBlock, ParameterInterface.parameter, SwcInternalBehavior.perInstance Parameter, SwcInternalBehavior.sharedParameter			
Attribute	Type	Mult.	Kind	Note
initValue	ValueSpecification	0..1	aggr	Specifies initial value(s) of the ParameterDataPrototype

Table A.690: ParameterDataPrototype

Class	ParameterInAtomicSWCTypeInstanceRef			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::DataElements::InstanceRefs Usage			
Note	This class implements an instance reference which can be applied for variables as well as for parameters.			
Base	*ARObject*, *AtpInstanceRef*			
Aggregated by	AutosarParameterRef.autosarParameter			
Attribute	Type	Mult.	Kind	Note
base	AtomicSwComponent Type	0..1	ref	**Stereotypes:** atpDerived **Tags:**xml.sequenceOffset=10
contextData Prototype (ordered)	ApplicationComposite ElementDataPrototype	*	ref	This ist the context in a compositeDataType. **Tags:**xml.sequenceOffset=40
portPrototype	PortPrototype	0..1	ref	This is the port providing the variable or the entry point to the variable structure. **Tags:**xml.sequenceOffset=20
rootParameter DataPrototype	DataPrototype	0..1	ref	This represents the entry point for references into a CompositeDataType. **Tags:**xml.sequenceOffset=30

▽

Collection of constraints on AUTOSAR M1 models
AUTOSAR CP R22-11

△

Class	ParameterInAtomicSWCTypeInstanceRef			
targetData Prototype	DataPrototype	0..1	ref	This is the target parameter element. Note that this must be nested in ParameterDataPrototype. The target must be one of ParameterDataPrototype, Application CompositeElementDataPrototype. **Tags:**xml.sequenceOffset=50

Table A.691: ParameterInAtomicSWCTypeInstanceRef

Class	ParameterInterface			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	A parameter interface declares a number of parameter and characteristic values to be exchanged between parameter components and software components. **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**
parameter	ParameterData Prototype	*	aggr	The ParameterDataPrototype of this ParameterInterface.

Table A.692: ParameterInterface

Class	ParameterPortAnnotation			
Package	M2::AUTOSARTemplates::SWComponentTemplate::ApplicationAttributes			
Note	Annotation to a port used for calibration regarding a certain ParameterDataPrototype.			
Base	*ARObject*, *GeneralAnnotation*			
Aggregated by	*PortPrototype*.parameterPortAnnotation			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
parameter	ParameterData Prototype	0..1	ref	The instance of annotated ParameterDataPrototype.

Table A.693: ParameterPortAnnotation

Class	ParameterProvideComSpec			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	"Communication" specification that applies to parameters on the provided side of a connection.			
Base	*ARObject*, *PPortComSpec*			
Aggregated by	*AbstractProvidedPortPrototype*.providedComSpec, PortPrototypeBlueprint.providedComSpec			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
initValue	ValueSpecification	0..1	aggr	The initial value applicable for the corresponding ParameterDataPrototype.
parameter	ParameterData Prototype	0..1	ref	The ParameterDataPrototype to which the Parameter ComSpec applies.

Table A.694: ParameterProvideComSpec

Class	ParameterRequireComSpec
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication
Note	"Communication" specification that applies to parameters on the required side of a connection.
Base	*ARObject*, *RPortComSpec*
Aggregated by	*AbstractRequiredPortPrototype*.requiredComSpec, PortPrototypeBlueprint.requiredComSpec

Attribute	Type	Mult.	Kind	Note
initValue	ValueSpecification	0..1	aggr	The initial value applicable for the corresponding ParameterDataPrototype.
parameter	ParameterData Prototype	0..1	ref	The ParameterDataPrototype to which the Parameter RequireComSpec applies.

Table A.695: ParameterRequireComSpec

Class	ParameterSwComponentType
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components
Note	The ParameterSwComponentType defines parameters and characteristic values accessible via provided Ports. The provided values are the same for all connected SwComponentPrototypes **Tags:**atp.recommendedPackage=SwComponentTypes
Base	*ARElement*, ARObject, *AtpBlueprint*, AtpBlueprintable, *AtpClassifier*, *AtpType*, CollectableElement, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SwComponentType*
Aggregated by	ARPackage.element

Attribute	Type	Mult.	Kind	Note
constant Mapping	ConstantSpecification MappingSet	*	ref	Reference to the ConstantSpecificationMapping to be applied for the particular ParameterSwComponentType **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=constantMapping
dataType Mapping	DataTypeMappingSet	*	ref	Reference to the DataTypeMapping to be applied for the particular ParameterSwComponentType **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=dataTypeMapping
instantiation DataDefProps	InstantiationDataDef Props	*	aggr	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 PortPrototypes **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=instantiationDataDefProps, instantiationData DefProps.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table A.696: ParameterSwComponentType

Class	PassThroughSwConnector
Package	M2::AUTOSARTemplates::SWComponentTemplate::Composition
Note	This kind of SwConnector can be used inside a CompositionSwComponentType to connect two delegation PortPrototypes.
Base	*ARObject*, *AtpClassifier*, AtpFeature, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *SwConnector*
Aggregated by	*AtpClassifier*.atpFeature, CompositionSwComponentType.connector

▽

Document ID 635: AUTOSAR_TR_ClassicAutosarModelConstraints

△

Class	**PassThroughSwConnector**			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
providedOuter Port	AbstractProvidedPort Prototype	0..1	ref	This represents the provided outer delegation Port Prototype of the PassThroughSwConnector.
requiredOuter Port	AbstractRequiredPort Prototype	0..1	ref	This represents the required outer delegation Port Prototype of the PassThroughSwConnector.

Table A.697: PassThroughSwConnector

Class	**Pdu** (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Collection of all Pdus that can be routed through a bus interface.			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Subclasses	GeneralPurposePdu, *IPdu*, NmPdu, UserDefinedPdu			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
hasDynamic Length	Boolean	0..1	attr	This attribute defines whether the Pdu has dynamic length (true) or not (false). Please note that the usage of this attribute is restricted by [constr_3448].
length	UnlimitedInteger	0..1	attr	Pdu length in bytes. In case of dynamic length IPdus (containing a dynamical length signal), this value indicates the maximum data length. It should be noted that in former AUTOSAR releases (Rel 2.1, Rel 3.0, Rel 3.1, Rel 4.0 Rev. 1) this parameter was defined in bits. The Pdu length of zero bytes is allowed.

Table A.698: Pdu

Class	**PduActivationRoutingGroup**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances			
Note	Group of Pdus that can be activated or deactivated for transmission over a socket connection.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AbstractServiceInstance*.methodActivationRoutingGroup, ConsumedEventGroup.pduActivationRouting Group, EventHandler.pduActivationRoutingGroup			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
eventGroup ControlType	EventGroupControlType Enum	0..1	attr	This attribute defines the type of a RoutingGroup. There are RoutingGroups that activate the data path for unicast or multicast events of an event group. And there are RoutingGroups that activate the data path for initial events that are triggered, namely events that are sent out on the server side after a client got subscribed. Please note that this attribute is only valid for event communication (Sender Receiver communication) and shall be omitted in MethodActivationRoutingGroups.
iPduIdentifier Tcp	SoConIPduIdentifier	*	ref	PduIdentifiers assigned for transmission over Tcp in case that the referencing PduActivationRoutingGroup is activated.
iPduIdentifier Udp	SoConIPduIdentifier	*	ref	PduIdentifiers assigned for transmission over Udp in case that the referencing PduActivationRoutingGroup is activated.

Table A.699: PduActivationRoutingGroup

Class	<<atpPrototype>> **PduToFrameMapping**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	A PduToFrameMapping defines the composition of Pdus in each frame.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*Frame*.pduToFrameMapping			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
packingByte Order	ByteOrderEnum	1	attr	This attribute defines the order of the bytes of the Pdu and the packing into the Frame. Please consider that [constr_3246] and [constr_3222] are restricting the usage of this attribute.
pdu	Pdu	1	ref	Reference to a I-Pdu, N-Pdu or NmPdu that is transmitted in the Frame.
startPosition	Integer	1	attr	This attribute describes the bitposition of a Pdu within a Frame.
				Please note that the absolute position of the Pdu in the Frame is determined by the definition of the packingByte Order attribute. If Big Endian is specified, the start position indicates the bit position of the most significant bit in the Frame. If Little Endian is specified, the start position indicates the bit position of the least significant bit in the Frame. The bit counting in byte 0 starts with bit 0 (least significant bit). The most significant bit in byte 0 is bit 7.
				The Pdus are byte aligned in a Frame and only the values 0, 8, 16, 24,... (for little endian) and 7, 15, 23, ... (for big endian) are allowed.
update IndicationBit Position	Integer	0..1	attr	Indication to the receivers that the corresponding Pdu was updated by the sender. This attribute describes the position of the update bit in the frame that aggregates this PDUToFrameMapping. Length is always one bit.
				Note that the exact bit position of the updateIndicationBit Position is linked to the value of the attribute packingByte Order because the method of finding the bit position is different for the values mostSignificantByteFirst and most SignificantByteLast. This means that if the value of packingByteOrder is changed while the value of update IndicationBitPosition remains unchanged the exact bit position of updateIndicationBitPosition within the enclosing Frame still undergoes a change.
				This attribute denotes the least significant bit for "Little Endian" and the most significant bit for "Big Endian" packed signals within the IPdu (see the description of the packingByteOrder attribute). In AUTOSAR the bit counting is always set to "sawtooth" and the bit order is set to "Decreasing". The bit counting in byte 0 starts with bit 0 (least significant bit). The most significant bit in byte 0 is bit 7.

Table A.700: PduToFrameMapping

Class	**PduTriggering**
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication
Note	The PduTriggering describes on which channel the IPdu is transmitted. The Pdu routing by the PduR is only allowed for subclasses of IPdu.
	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.
	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.

$\triangledown$

△

Class	PduTriggering			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*PhysicalChannel*.pduTriggering			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
iPdu	Pdu	1	ref	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. 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.
iPduPort	IPduPort	*	ref	References to the IPduPort on every ECU of the system which sends and/or receives the I-PDU. References for both the sender and the receiver side shall be included when the system is completely defined.
iSignal Triggering	ISignalTriggering	*	ref	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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=iSignalTriggering.iSignalTriggering, iSignal Triggering.variationPoint.shortLabel vh.latestBindingTime=postBuild
secOcCrypto Mapping	SecOcCryptoService Mapping	0..1	ref	This reference identifies the crypto profile applicable to the usage (send, receive) of the also referenced Secured IPdu. Obviously, this reference is only applicable if the Pdutriggering also references a SecuredIPdu in the role i Pdu.
triggerIPduSend Condition	TriggerIPduSend Condition	*	aggr	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.

Table A.701: PduTriggering

Class	PerInstanceMemory			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::PerInstanceMemory			
Note	Defines a 'C' typed memory-block that needs to be available for each instance of the SW-component. This is typically only useful if supportsMultipleInstantiation is set to "true" or if the software-component defines NVRAM access via permanent blocks.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, SwcInternalBehavior.perInstanceMemory			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
initValue	String	0..1	attr	Specifies initial value(s) of the PerInstanceMemory
swDataDef Props	SwDataDefProps	0..1	aggr	This represents the ability to to allocate RAM at specific memory sections, for example, to support the RAM Block recovery strategy by mapping to uninitialized RAM. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=swDataDefProps
type	CIdentifier	0..1	attr	The name of the "C"-type
typeDefinition	String	0..1	attr	A definition of the type with the syntax of a 'C' typedef.

Table A.702: PerInstanceMemory

Class	PerInstanceMemorySize			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcImplementation			
Note	Resources needed by the allocation of PerInstanceMemory for each SWC instance. Note that these resources are not covered by an ObjectFileSection, because they are supposed to be allocated by the RTE.			
Base	ARObject			
Aggregated by	SwcImplementation.perInstanceMemorySize			
Attribute	Type	Mult.	Kind	Note
alignment	PositiveInteger	0..1	attr	Required alignment (1,2,4,...) of the referenced Per InstanceMemory. Unit: byte.
perInstance Memory	PerInstanceMemory	0..1	ref	This represents the referenced PerInstanceMemory.
size	PositiveInteger	0..1	attr	Size (in bytes) of the reference perInstanceMemory. The aggregation of PerInstanceMemorySize is subject to variability with the purpose to support variability in the software components implementations. Different algorithms in the implementation might require a different PerInstanceMemorySize. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime

Table A.703: PerInstanceMemorySize

Class	PeriodicEventTriggering			
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint::EventTriggeringConstraint			
Note	The PeriodicEventTriggering describes the behavior of an event with a strict periodic occurrence pattern, given by the period attribute. Additionally, it is possible to soften the strictness of the periodic occurrence behavior by specifying a jitter, so that there can be a deviation from the period up to the size of the jitter.			
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 jitter of the periodic event occurrence. **Tags:**xml.sequenceOffset=20
minimumInter ArrivalTime	MultidimensionalTime	0..1	aggr	The minimum time distance between two consecutive occurrences of the associated event. **Tags:**xml.sequenceOffset=10
period	MultidimensionalTime	0..1	aggr	The period of the event occurrence. **Tags:**xml.sequenceOffset=30

Table A.704: PeriodicEventTriggering

Class	PhysConstrs			
Package	M2::MSR::AsamHdo::Constraints::GlobalConstraints			
Note	This meta-class represents the ability to express physical constraints. Therefore it has (in opposite to InternalConstrs) a reference to a Unit.			
Base	ARObject			
Aggregated by	DataConstrRule.physConstrs			
Attribute	Type	Mult.	Kind	Note

▽

△

Class	PhysConstrs			
lowerLimit	Limit	0..1	attr	This specifies the lower limit of the constraint. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=20
maxDiff	Numerical	0..1	attr	Maximum difference that is permitted between two consecutive values if the constraint is applied to an axis. **Tags:**xml.sequenceOffset=60
maxGradient	Numerical	0..1	attr	This element specifies the maximum slope that may be used in curves and maps. **Tags:**xml.sequenceOffset=50
monotony	MonotonyEnum	0..1	attr	This specifies the monotony constraints on the data object. Note that this applies only to curves and maps. **Tags:**xml.sequenceOffset=70
scaleConstr (ordered)	ScaleConstr	*	aggr	This is one particular scale which contributes to the data constraints. **Tags:** xml.roleElement=true xml.roleWrapperElement=true xml.sequenceOffset=40 xml.typeElement=false xml.typeWrapperElement=false
unit	Unit	0..1	ref	This is the unit to which the physical constraints relate to. In particular, it is the physical unit of the specified limits. **Tags:**xml.sequenceOffset=80
upperLimit	Limit	0..1	attr	This specifies the upper limit of the constraint. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=30

Table A.705: PhysConstrs

Class	PhysicalChannel (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology			
Note	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. 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.#			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Subclasses	AbstractCanPhysicalChannel, EthernetPhysicalChannel, FlexrayPhysicalChannel, LinPhysicalChannel, UserDefinedPhysicalChannel			
Aggregated by	CommunicationCluster.physicalChannel			
Attribute	Type	Mult.	Kind	Note

▽

△

Class	*PhysicalChannel* (abstract)			
comm Connector	Communication Connector	*	ref	Reference to the ECUInstance via a Communication Connector to which the channel is connected. atpVariation: Variable assignment of Physical Channels to different CommunicationConnectors is expressed with this variation. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=commConnector.communicationConnector, commConnector.variationPoint.shortLabel vh.latestBindingTime=postBuild
frameTriggering	FrameTriggering	*	aggr	One frame triggering is defined for exactly one channel. Channels may have assigned an arbitrary number of frame triggerings. atpVariation: If signals/PDUs/frames are variable, the corresponding triggerings shall be variable, too. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=frameTriggering.shortName, frame Triggering.variationPoint.shortLabel vh.latestBindingTime=postBuild
iSignal Triggering	ISignalTriggering	*	aggr	One ISignalTriggering is defined for exactly one channel. Channels may have assigned an arbitrary number of ISignaltriggerings. atpVariation: If signals/PDUs/frames are variable, the corresponding triggerings shall be variable, too. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=iSignalTriggering.shortName, iSignal Triggering.variationPoint.shortLabel vh.latestBindingTime=postBuild
managed Physical Channel	PhysicalChannel	*	ref	Reference between a channel with role managing channel and a channel with role managed channel.
pduTriggering	PduTriggering	*	aggr	One PduTriggering is defined for exactly one channel. Channels may have assigned an arbitrary number of I-Pdu triggerings. atpVariation: If signals/PDUs/frames are variable, the corresponding triggerings shall be variable, too. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=pduTriggering.shortName, pdu Triggering.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.706: PhysicalChannel

Class	PhysicalDimension			
Package	M2::MSR::AsamHdo::Units			
Note	This class represents a physical dimension. If the physical dimension of two units is identical, then a conversion between them is possible. The conversion between units is related to the definition of the physical dimension.			
	Note that the equivalence of the exponents does not per se define the convertibility. For example Energy and Torque share the same exponents (Nm).			
	Please note further the value of an exponent does not necessarily have to be an integer number. It is also possible that the value yields a rational number, e.g. to compute the square root of a given physical quantity. In this case the exponent value would be a rational number where the numerator value is 1 and the denominator value is 2.			
	Tags:atp.recommendedPackage=PhysicalDimensions			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
currentExp	Numerical	0..1	attr	This attribute represents the exponent of the physical dimension "electric current".
				Tags:xml.sequenceOffset=50
lengthExp	Numerical	0..1	attr	The exponent of the physical dimension "length".
				Tags:xml.sequenceOffset=20
luminous IntensityExp	Numerical	0..1	attr	The exponent of the physical dimension "luminous intensity".
				Tags:xml.sequenceOffset=80
massExp	Numerical	0..1	attr	The exponent of the physical dimension "mass".
				Tags:xml.sequenceOffset=30
molarAmount Exp	Numerical	0..1	attr	The exponent of the physical dimension "quantity of substance".
				Tags:xml.sequenceOffset=70
temperatureExp	Numerical	0..1	attr	The exponent of the physical dimension "temperature".
				Tags:xml.sequenceOffset=60
timeExp	Numerical	0..1	attr	The exponent of the physical dimension "time".
				Tags:xml.sequenceOffset=40

Table A.707: PhysicalDimension

Class	PhysicalDimensionMapping			
Package	M2::MSR::AsamHdo::Units			
Note	This class represents a specific mapping between two PhysicalDimensions.			
Base	*ARObject*			
Aggregated by	PhysicalDimensionMappingSet.physicalDimensionMapping			
Attribute	Type	Mult.	Kind	Note
firstPhysical Dimension	PhysicalDimension	0..1	ref	This represents the first PhysicalDimension of the enclosing PhysicalDimensionMapping.
secondPhysical Dimension	PhysicalDimension	0..1	ref	This represents the first PhysicalDimension of the enclosing PhysicalDimensionMapping.

Table A.708: PhysicalDimensionMapping

Class	PlcaProps
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology
Note	This meta-class allows to configure the PLCA (Physical Layer Collision Avoidance) in case 10-BASE-T1S Ethernet is used and PLCA is enabled on the CouplingPort (PHY).
Base	ARObject
Aggregated by	CouplingPort.plcaProps

Attribute	Type	Mult.	Kind	Note
plcaLocalNode Id	PositiveInteger	0..1	attr	This attribute defines the node ID when the PLCA mode for 10BASE-T1S is used.
plcaMaxBurst Count	PositiveInteger	0..1	attr	Defines maximum packets allowed to be transmitted within a TO. This configuration can be different from one ECU to another within the PLCA mixed segment.
plcaMaxBurst Timer	PositiveInteger	0..1	attr	Limits the burst frames in bit time. This configuration can be different from one ECU to another within the PLCA mixed segment. For PLCA burst mode to work properly this timer should be set greater than one IPG.

Table A.709: PlcaProps

Enumeration	PncGatewayTypeEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology
Note	Defines the PncGateway roles.
Aggregated by	CommunicationConnector.pncGatewayType

Literal	Description
active	The active PncGateway functionality shall be performed **Tags:**atp.EnumerationLiteralIndex=0
none	No PncGateway functionality shall be performed **Tags:**atp.EnumerationLiteralIndex=1
passive	The passive PncGateway functionality shall be performed **Tags:**atp.EnumerationLiteralIndex=2

Table A.710: PncGatewayTypeEnum

Class	PncMapping
Package	M2::AUTOSARTemplates::SystemTemplate::PncMapping
Note	Describes a mapping between one or several Virtual Function Clusters onto Partial Network Clusters. A Virtual Function Cluster is realized by a PortGroup. A Partial Network Cluster is realized by one or more IPduGroups.
Base	ARObject, Describable
Aggregated by	SystemMapping.pncMapping

Attribute	Type	Mult.	Kind	Note
dynamicPnc MappingPdu Group	ISignalIPduGroup	*	ref	Reference to an ISignalIPduGroup that allows mapping of this PNC without statically mapping this PNC directly to a channel. This is needed to describe dynamic PNCs that can be learned only at run-time and which have also a relation to an ISignalIPduGroup. **Tags:**atp.Status=draft
ident	PncMappingIdent	0..1	aggr	This adds the ability to become referrable to PncMapping.
physical Channel	PhysicalChannel	*	ref	This reference maps the partial network to a communication channel.

▽

△

Class	PncMapping			
pncConsumed Provided ServiceInstance Group	ConsumedProvided ServiceInstanceGroup	*	ref	ConsumedProvidedServiceInstanceGroup used in a Partial Network Cluster. This reference is optional, since this could be used for starting and stopping Consumed ProvidedServiceInstanceGroup according the requested partial network, but is not necessarily needed. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=pncConsumedProvidedServiceInstance Group.consumedProvidedServiceInstanceGroup, pnc ConsumedProvidedServiceInstanceGroup.variation Point.shortLabel vh.latestBindingTime=postBuild
pncGroup	ISignalIPduGroup	*	ref	IPduGroup participating in a Partial Network Cluster. This reference is optional in case an ecu extract has only indirect pnc access, i.e. ecu is not directly connected to a network which supports partial network.
pncIdentifier	PositiveInteger	1	attr	Identifer of the Partial Network Cluster. This number represents the absolute bit position of this Partial Network Cluster in the NM Pdu.
pncPdurGroup	PdurIPduGroup	*	ref	This reference maps the Partial Network Cluster to a set of PdurIpduGroups.
pncWakeup Enable	Boolean	0..1	attr	If this parameter is available and set to true then this PNC will be woken up as soon as a channel wakeup occurs on a channel where this PNC is assigned to. This is ensured by adding this PNC to the corresponding channel wakeup sources during upstream mapping.
relevantFor DynamicPnc Mapping	EcuInstance	*	ref	Reference to a PNC Gateway ECU for PNCs which do not have a static channel mapping. This is needed to describe dynamic PNCs that can be learned only at run-time and which have no relation to an ISignalIPdu Group. **Tags:**atp.Status=draft
shortLabel	Identifier	0..1	attr	This attribute specifies an identifying shortName for the PncMapping. It shall be unique in the System scope.
vfc	PortGroup	*	iref	Virtual Function Cluster to be mapped onto a Partial Network Cluster. This reference is optional in case that the System Description doesn't use a complete Software Component Description (VFB View). This supports the inclusion of legacy systems. **InstanceRef implemented by:**PortGroupInSystem InstanceRef
wakeupFrame	FrameTriggering	*	ref	Reference to collection of FrameTriggerings that are used for the wakeup of this PNC (Application Frames or Nm Frames can be used). This reference is only valid if this EcuExtract represents an ECU which has direct PNC access, i.e. ECU is directly connected to a network which supports partial network.

Table A.711: PncMapping

Class	PncMappingIdent	
Package	M2::AUTOSARTemplates::SystemTemplate::PncMapping	
Note	This meta-class is created to add the ability to become the target of a reference to the non-Referrable PncMapping.	
Base	*ARObject*, *Referrable*	

▽

△

Class	PncMappingIdent			
Aggregated by	PncMapping.ident			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.712: PncMappingIdent

Class	PortAPIOption			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::PortAPIOptions			
Note	Options how to generate the signatures of calls for an AtomicSwComponentType in order to communicate over a PortPrototype (for calls into a RunnableEntity as well as for calls from a Runnable Entity to the PortPrototype).			
Base	ARObject			
Aggregated by	SwcInternalBehavior.portAPIOption			
Attribute	Type	Mult.	Kind	Note
enableTake Address	Boolean	0..1	attr	If set to true, the software-component is able to use the API reference for deriving a pointer to an object.
errorHandling	DataTransformation ErrorHandlingEnum	0..1	attr	This specifies whether a RunnableEntity accessing a Port Prototype that is referenced by this PortAPIOption shall specifically handle transformer errors or not.
indirectAPI	Boolean	0..1	attr	If set to true this attribute specifies an "indirect API" to be generated for the associated port which means that the software-component is able to access the actions on a port via a pointer to an object representing a port. This allows e.g. iterating over ports in a loop. This option has no effect for PPortPrototypes of client/server interfaces.
port	PortPrototype	0..1	ref	The option is valid for generated functions related to communication over this port
portArgValue (ordered)	PortDefinedArgument Value	*	aggr	An argument value defined by this port.
supported Feature	SwcSupportedFeature	*	aggr	This collection specifies which features are supported by the RunnableEntitys which access a PortPrototype that it referenced by this PortAPIOption.
transformer Status Forwarding	DataTransformation StatusForwardingEnum	0..1	attr	This attribute specifies whether a RunnableEntity accessing a PortPrototype that is referenced by this Port APIOption shall be able to forward a status code to the transformer chain.

Table A.713: PortAPIOption

Class	PortDefinedArgumentValue			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::PortAPIOptions			
Note	A PortDefinedArgumentValue is passed to a RunnableEntity dealing with the ClientServerOperations provided by a given PortPrototype. Note that this is restricted to PPortPrototypes of a ClientServer Interface.			
Base	ARObject			
Aggregated by	PortAPIOption.portArgValue			
Attribute	Type	Mult.	Kind	Note
value	ValueSpecification	0..1	aggr	Specifies the actual value.

▽

△

Class	PortDefinedArgumentValue			
valueType	ImplementationData Type	0..1	tref	The implementation type of this argument value. It should not be composite type or a pointer. **Stereotypes:** isOfType

Table A.714: PortDefinedArgumentValue

Class	PortElementToCommunicationResourceMapping			
Package	M2::AUTOSARTemplates::SystemTemplate			
Note	This meta class maps a communication resource to CP Software Clusters. In this case the kind of Port Prototype specified whether the Software Cluster has to provide or to require the resource. **Tags:**atp.Status=draft			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	CpSoftwareClusterMappingSet.portElementToComResourceMapping, SystemMapping.portElementToComResourceMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
clientServer Operation	ClientServerOperation	0..1	iref	ClientServerOperation instance qualifying the communication resource **Tags:**atp.Status=draft **InstanceRef implemented by:**OperationInSystem InstanceRef
communication Resource	CpSoftwareCluster Communication Resource	0..1	ref	Communication resource for which the mapping applies. **Tags:**atp.Status=draft
mode Declaration GroupPrototype	ModeDeclarationGroup Prototype	0..1	iref	ModeDeclarationGroupPrototype instance qualifying the communication resource **Tags:**atp.Status=draft **InstanceRef implemented by:**ModeDeclarationGroup PrototypeInSystemInstanceRef
parameterData Prototype	ParameterData Prototype	0..1	iref	ParameterDataPrototype instance qualifying the communication resource. **Tags:**atp.Status=draft **InstanceRef implemented by:**ParameterDataPrototype InSystemInstanceRef
trigger	Trigger	0..1	iref	Trigger instance qualifying the communication resource. **Tags:**atp.Status=draft **InstanceRef implemented by:**TriggerInSystemInstance Ref
variableData Prototype	VariableDataPrototype	0..1	iref	VariableDataPrototype instance qualifying the communication resource **Tags:**atp.Status=draft **InstanceRef implemented by:**VariableDataPrototypeIn SystemInstanceRef

Table A.715: PortElementToCommunicationResourceMapping

Class	PortGroup			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components			
Note	Group of ports which share a common functionality			
	, e.g. need specific network resources. This information shall be available on the VFB level in order to delegate it properly via compositions. When propagated into the ECU extract, this information is used as input for the configuration of Services like the Communication Manager.			
	A PortGroup is defined locally in a component (which can be a composition) and refers to the "outer" ports belonging to the group as well as to the "inner" groups which propagate this group into the components which are part of a composition. A PortGroup within an atomic SWC cannot be linked to inner groups.			
Base	ARObject, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	AtpClassifier.atpFeature, SwComponentType.portGroup			
Attribute	Type	Mult.	Kind	Note
innerGroup	PortGroup	*	iref	Links a PortGroup in a composition to another PortGroup, that is defined in a component which is part of this CompositionSwComponentType.
				InstanceRef implemented by:InnerPortGroupIn CompositionInstanceRef
outerPort	PortPrototype	*	ref	Outer PortPrototype of this AtomicSwComponentType which belongs to the group. A port can belong to several groups or to no group at all.
				Stereotypes: atpSplitable; atpVariation Tags: atp.Splitkey=outerPort.portPrototype, outerPort.variation Point.shortLabel vh.latestBindingTime=preCompileTime

Table A.716: PortGroup

Class	PortInterface (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	ARElement, ARObject, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Subclasses	ClientServerInterface, DataInterface, ModeSwitchInterface, TriggerInterface			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
isService	Boolean	0..1	attr	This flag is set if the PortInterface is to be used for communication between an
				• ApplicationSwComponentType or
				• ServiceProxySwComponentType or
				• SensorActuatorSwComponentType or
				• ComplexDeviceDriverSwComponentType
				• ServiceSwComponentType
				• EcuAbstractionSwComponentType
				and a ServiceSwComponentType (namely an AUTOSAR Service) located on the same ECU. Otherwise the flag is not set.
				Stereotypes: atpVariation Tags:vh.latestBindingTime=blueprintDerivationTime

△

Class	PortInterface (abstract)			
serviceKind	ServiceProviderEnum	0..1	attr	This attribute provides further details about the nature of the applied service.

Table A.717: PortInterface

Class	PortInterfaceMapping (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	Specifies one PortInterfaceMapping to support the connection of Ports typed by two different Port Interfaces with PortInterface elements having unequal names and/or unequal semantic (resolution or range).			
Base	ARObject, AtpBlueprint, AtpBlueprintable, Identifiable, MultilanguageReferrable, Referrable			
Subclasses	ClientServerInterfaceMapping, ModeInterfaceMapping, TriggerInterfaceMapping, VariableAndParameterInterfaceMapping			
Aggregated by	PortInterfaceMappingSet.portInterfaceMapping			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.718: PortInterfaceMapping

Class	PortPrototype (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components			
Note	Base class for the ports of an AUTOSAR software component. The aggregation of PortPrototypes is subject to variability with the purpose to support the conditional existence of ports.			
Base	ARObject, AtpBlueprintable, AtpFeature, AtpPrototype, Identifiable, MultilanguageReferrable, Referrable			
Subclasses	AbstractProvidedPortPrototype, AbstractRequiredPortPrototype			
Aggregated by	AtpClassifier.atpFeature, SwComponentType.port			
Attribute	Type	Mult.	Kind	Note
clientServer Annotation	ClientServerAnnotation	*	aggr	Annotation of this PortPrototype with respect to client/server communication.
delegatedPort Annotation	DelegatedPortAnnotation	0..1	aggr	Annotations on this delegated port.
ioHwAbstraction Server Annotation	IoHwAbstractionServerAnnotation	*	aggr	Annotations on this IO Hardware Abstraction port.
modePort Annotation	ModePortAnnotation	*	aggr	Annotations on this mode port.
nvDataPort Annotation	NvDataPortAnnotation	*	aggr	Annotations on this non voilatile data port.
parameterPort Annotation	ParameterPortAnnotation	*	aggr	Annotations on this parameter port.
senderReceiver Annotation	SenderReceiverAnnotation	*	aggr	Collection of annotations of this ports sender/receiver communication.
triggerPort Annotation	TriggerPortAnnotation	*	aggr	Annotations on this trigger port.

Table A.719: PortPrototype

Class	PortPrototypeBlueprint			
Package	M2::AUTOSARTemplates::CommonStructure::StandardizationTemplate::BlueprintDedicated::Port ProtoypeBlueprint			
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*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
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 A.720: PortPrototypeBlueprint

Primitive	PositiveUnlimitedInteger
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::PrimitiveTypes
Note	This is a positive unlimited integer which can be denoted in decimal, binary, octal and hexadecimal. **Tags:** xml.xsd.customType=POSITIVE-UNLIMITED-INTEGER xml.xsd.pattern=0|[\+]?[1-9][0-9]*|0[xX][0-9a-fA-F]+|0[bB][0-1]+|0[0-7]+ xml.xsd.type=string

Table A.721: PositiveUnlimitedInteger

Class	PostBuildVariantCondition			
Package	M2::AUTOSARTemplates::GenericStructure::VariantHandling			
Note	This class specifies the value which shall be assigned to a particular variant criterion in order to bind the variation point. If multiple criterion/value pairs are specified, they shall all match to bind the variation point. In other words binding can be represented by (criterion1 == value1) && (condition2 == value2) ...			
Base	*ARObject*			
Aggregated by	VariationPoint.postBuildVariantCondition, VariationPointProxy.postBuildVariantCondition			
Attribute	Type	Mult.	Kind	Note
matching Criterion	PostBuildVariant Criterion	1	ref	This is the criterion which needs to match the value in order to make the PostbuildVariantCondition to be true.
value	Integer	1	attr	This is the particular value of the post-build variant criterion. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime

Table A.722: PostBuildVariantCondition

Class	PredefinedVariant			
Package	M2::AUTOSARTemplates::GenericStructure::VariantHandling			
Note	This specifies one predefined variant. It is characterized by the union of all system constant values and post-build variant criterion values aggregated within all referenced system constant value sets and post build variant criterion value sets plus the value sets of the included variants. **Tags:**atp.recommendedPackage=PredefinedVariants			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
includedVariant	PredefinedVariant	*	ref	The associated variants are considered part of this PredefinedVariant. This means the settings of the included variants are included in the settings of the referencing PredefinedVariant. Nevertheless the included variants might be included in several predefined variants.
postBuildVariant CriterionValue Set	PostBuildVariant CriterionValueSet	*	ref	This is the postBuildVariantCriterionValueSet contributing to the predefinded variant.
sw Systemconstant ValueSet	SwSystemconstant ValueSet	*	ref	This ist the set of Systemconstant Values contributing to the predefined variant.

Table A.723: PredefinedVariant

Class	PrimitiveAttributeCondition			
Package	M2::AUTOSARTemplates::CommonStructure::StandardizationTemplate::DataExchangePoint::Data FormatTailoring			
Note	The PrimitiveAttributeCondition evaluates to true, if the referenced primitive attribute is accepted by all rules of this condition.			
Base	*ARObject*, *AbstractCondition*, *AbstractMultiplicityRestriction*, *AbstractValueRestriction*, *AttributeCondition*			
Aggregated by	ClassContentConditional.condition, InvertCondition.condition			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
attribute	PrimitiveAttribute Tailoring	1	ref	The primitive attribute that has to be accepted by the restrictions of this PrimitiveAttributeCondition

Table A.724: PrimitiveAttributeCondition

Class	PrimitiveAttributeTailoring			
Package	M2::AUTOSARTemplates::CommonStructure::StandardizationTemplate::DataExchangePoint::Data FormatTailoring			
Note	Tailoring of primitive attributes. Primitive attributes are attributes that have a type which is marked by the stereotype <<primitive>> or <<enumeration>>			
Base	*ARObject*, *AttributeTailoring*, *DataFormatElementReference*, *DataFormatElementScope*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *SpecElementReference*, *SpecElementScope*			
Aggregated by	ClassContentConditional.attributeTailoring, PrimitiveAttributeTailoring.subAttributeTailoring			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
defaultValue Handling	DefaultValueApplication StrategyEnum	0..1	attr	Specification of how to handle AUTOSAR defined default values.
subAttribute Tailoring	PrimitiveAttribute Tailoring	*	aggr	Tailors the attribute of a <<primitive>> data type.
valueRestriction	ValueRestrictionWith Severity	0..1	aggr	The restriction of the attribute value.

Table A.725: PrimitiveAttributeTailoring

Class	**ProvidedServiceInstance**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances			
Note	Service instances that are provided by the ECU that is connected via the ApplicationEndpoint to a CommunicationConnector.			
Base	*ARObject*, *AbstractServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ApplicationEndpoint.providedServiceInstance, ServiceInstanceCollectionSet.serviceInstance			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
autoAvailable	Boolean	0..1	attr	Defines that this ProvidedServiceInstance shall be offered by the service discovery at ECU start.
eventHandler	EventHandler	*	aggr	Collection of event groups provided by the Provided ServiceInstance **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=eventHandler.shortName, event Handler.variationPoint.shortLabel vh.latestBindingTime=postBuild
instance Identifier	PositiveInteger	0..1	attr	Instance identifier. Can be used for e.g. service discovery to identify the instance of the service.
loadBalancing Priority	PositiveInteger	0..1	attr	Defines the value to be used for load balancing priority in the service offer. Lower value means higher priority.
loadBalancing Weight	PositiveInteger	0..1	attr	Defines the value to be used for load balancing weight in the service offer. Higher value means higher probability to be chosen.
localUnicast Address	ApplicationEndpoint	0..2	ref	The local address over which the PSI is provided (udp, tcp or both). **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=localUnicastAddress.applicationEndpoint, localUnicastAddress.variationPoint.shortLabel vh.latestBindingTime=postBuild
minorVersion	PositiveInteger	0..1	attr	Minor Version of the Service that is provided by this ProvidedServiceInstance.
priority	PositiveInteger	0..1	attr	Defines the frame priority where values from 0 (best effort) to 7 (highest) are allowed.
remoteMulticast Subscription Address	ApplicationEndpoint	*	ref	This reference defines the remote multicast subscribed addresses of service consumers. This reference shall ONLY be used if the remote address of the clients is determined from the configuration and not at runtime. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=remoteMulticastSubscription Address.applicationEndpoint, remoteMulticast SubscriptionAddress.variationPoint.shortLabel vh.latestBindingTime=postBuild
remoteUnicast Address	ApplicationEndpoint	*	ref	This reference defines the remote addresses of service consumers. This reference shall ONLY be used if the remote address of the clients is determined from the configuration and not at runtime. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=remoteUnicastAddress.applicationEndpoint, remoteUnicastAddress.variationPoint.shortLabel vh.latestBindingTime=postBuild
sdServerConfig	SdServerConfig	0..1	aggr	Service Discovery Server configuration. **Tags:**atp.Status=obsolete

$\bigtriangledown$

△

Class	ProvidedServiceInstance			
sdServerTimer Config	SomeipSdServer ServiceInstanceConfig	0..1	ref	Server specific configuration settings relevant for the SOME/IP service discovery. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=sdServerTimerConfig.someipSdServer ServiceInstanceConfig, sdServerTimerConfig.variation Point.shortLabel vh.latestBindingTime=postBuild
serviceIdentifier	PositiveInteger	0..1	attr	This attribute represents the ability to describe the SOME/ IP service ID that is offered.

Table A.726: ProvidedServiceInstance

Class	QueuedReceiverComSpec			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	Communication attributes specific to queued receiving.			
Base	ARObject, RPortComSpec, ReceiverComSpec			
Aggregated by	AbstractRequiredPortPrototype.requiredComSpec, PortPrototypeBlueprint.requiredComSpec			
Attribute	Type	Mult.	Kind	Note
queueLength	PositiveInteger	0..1	attr	Length of queue for received events.

Table A.727: QueuedReceiverComSpec

Class	QueuedSenderComSpec			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	Communication attributes specific to distribution of events (PPortPrototype, SenderReceiverInterface and dataElement carries an "event").			
Base	ARObject, PPortComSpec, SenderComSpec			
Aggregated by	AbstractProvidedPortPrototype.providedComSpec, PortPrototypeBlueprint.providedComSpec			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.728: QueuedSenderComSpec

Class	RPortComSpec (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	Communication attributes of a required PortPrototype. This class will contain attributes that are valid for all kinds of require-ports, independent of client-server or sender-receiver communication patterns.			
Base	ARObject			
Subclasses	ClientComSpec, ModeSwitchReceiverComSpec, NvRequireComSpec, ParameterRequireComSpec, ReceiverComSpec			
Aggregated by	AbstractRequiredPortPrototype.requiredComSpec, PortPrototypeBlueprint.requiredComSpec			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.729: RPortComSpec

Class	RPortPrototype			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components			
Note	Component port requiring a certain port interface.			
Base	ARObject, AbstractRequiredPortPrototype, AtpBlueprintable, AtpFeature, AtpPrototype, Identifiable, MultilanguageReferrable, PortPrototype, Referrable			
Aggregated by	AtpClassifier.atpFeature, SwComponentType.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 A.730: RPortPrototype

Class	RTEEvent (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents			
Note	Abstract base class for all RTE-related events			
Base	ARObject, AbstractEvent, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable, Multilanguage Referrable, Referrable			
Subclasses	AsynchronousServerCallReturnsEvent, BackgroundEvent, DataReceiveErrorEvent, DataReceivedEvent, DataSendCompletedEvent, DataWriteCompletedEvent, ExternalTriggerOccurredEvent, InitEvent, InternalTriggerOccurredEvent, ModeSwitchedAckEvent, OperationInvokedEvent, OsTaskExecutionEvent, SwcModeManagerErrorEvent, SwcModeSwitchEvent, TimingEvent, TransformerHardErrorEvent			
Aggregated by	AtpClassifier.atpFeature, SwcInternalBehavior.event			
Attribute	Type	Mult.	Kind	Note
disabledMode	ModeDeclaration	*	iref	Reference to the Modes that disable the Event. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=disabledMode.contextPort, disabled Mode.contextModeDeclarationGroupPrototype, disabled Mode.targetModeDeclaration **InstanceRef implemented by:**RModeInAtomicSwc InstanceRef
startOnEvent	RunnableEntity	0..1	ref	The referenced RunnableEntity starts when the corresponding RTEEvent is raised.

Table A.731: RTEEvent

Class	RapidPrototypingScenario			
Package	M2::AUTOSARTemplates::SWComponentTemplate::RPTScenario			
Note	This meta-class provides the ability to describe a Rapid Prototyping Scenario. Such a Rapid Prototyping Scenario consist out of two main aspects, the description of the byPassPoints and the relation to an rpt Hook. **Tags:**atp.recommendedPackage=RapidPrototypingScenarios			
Base	ARElement, ARObject, CollectableElement, Identifiable, MultilanguageReferrable, Packageable Element, Referrable			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
hostSystem	System	0..1	ref	System which describes the software components of the host ECU.

▽

△

Class	RapidPrototypingScenario			
rptContainer	RptContainer	*	aggr	Top-level rptContainer definitions of this specific rapid prototyping scenario. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=rptContainer.shortName, rptContainer.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
rptProfile	RptProfile	*	aggr	Defiens the applicable Rapid Prototyping profils which are especially defining the smbol of the service functions and the valid id range. The order of the RptProfiles determines the order of the service function invocation by RTE. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=rptProfile.shortName
rptSystem	System	0..1	ref	System which describes the rapid prototyping algorithm in the format of AUTOSAR Software Components. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=rptSystem

Table A.732: RapidPrototypingScenario

Class	ReceiverComSpec (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	Receiver-specific communication attributes (RPortPrototype typed by SenderReceiverInterface).			
Base	ARObject, RPortComSpec			
Subclasses	NonqueuedReceiverComSpec, QueuedReceiverComSpec			
Aggregated by	AbstractRequiredPortPrototype.requiredComSpec, PortPrototypeBlueprint.requiredComSpec			
Attribute	Type	Mult.	Kind	Note
composite Network Representation	CompositeNetwork Representation	*	aggr	This represents a CompositeNetworkRepresentation defined in the context of a ReceiverComSpec. The purpose of this aggregation is to be able to specify the network representation of leaf elements of Application CompositeDataTypes.
dataElement	AutosarDataPrototype	0..1	ref	Data element these attributes belong to.
handleOutOf Range	HandleOutOfRange Enum	0..1	attr	This attribute controls how values that are out of the specified range are handled according to the values of HandleOutOfRangeEnum.
handleOutOf RangeStatus	HandleOutOfRange StatusEnum	0..1	attr	Control the way how return values are created in case of an out-of-range situation.

▽

△

Class	*ReceiverComSpec* (abstract)			
maxDelta CounterInit	PositiveInteger	0..1	attr	Initial maximum allowed gap between two counter values of two consecutively received valid Data, i.e. how many subsequent lost data is accepted. For example, if the receiver gets Data with counter 1 and MaxDeltaCounter Init is 1, then at the next reception the receiver can accept Counters with values 2 and 3, but not 4.
				Note that if the receiver does not receive new Data at a consecutive read, then the receiver increments the tolerance by 1.
				Caveat: The E2E wrapper approach involves technologies that are not subjected to the AUTOSAR standard and is superseded by the superior E2E transformer approach (which is fully standardized by AUTOSAR). Hence, new projects (without legacy constraints due to carry-over parts) shall use the fully standardized E2E transformer approach.
				Stereotypes: atpVariation **Tags:**vh.latestBindingTime=preCompileTime
maxNoNewOr RepeatedData	PositiveInteger	0..1	attr	The maximum amount of missing or repeated Data which the receiver does not expect to exceed under normal communication conditions.
				Caveat: The E2E wrapper approach involves technologies that are not subjected to the AUTOSAR standard and is superseded by the superior E2E transformer approach (which is fully standardized by AUTOSAR). Hence, new projects (without legacy constraints due to carry-over parts) shall use the fully standardized E2E transformer approach.
network Representation	[SwDataDefProps]	0..1	aggr	A networkRepresentation is used to define how the data Element is mapped to a communication bus.
				Stereotypes: atpSplitable **Tags:**atp.Splitkey=networkRepresentation
receptionProps	[ReceptionComSpec Props]	0..1	aggr	"This aggregation represents the definition transmission props in the context of the enclosing ReceiverComSpec.
replaceWith	[VariableAccess]	0..1	aggr	This aggregation is used to identify the AutosarData Prototype to be taken for sourcing an external replacement in the out-of-range and invalidValue handling.
syncCounterInit	PositiveInteger	0..1	attr	Number of Data required for validating the consistency of the counter that shall be received with a valid counter (i.e. counter within the allowed lock-in range) after the detection of an unexpected behavior of a received counter.
				Caveat: The E2E wrapper approach involves technologies that are not subjected to the AUTOSAR standard and is superseded by the superior E2E transformer approach (which is fully standardized by AUTOSAR). Hence, new projects (without legacy constraints due to carry-over parts) shall use the fully standardized E2E transformer approach.
transformation ComSpecProps	TransformationCom SpecProps	*	aggr	This references the TransformationComSpecProps which define port-specific configuration for data transformation.

▽

△

Class	ReceiverComSpec (abstract)			
usesEndToEnd Protection	Boolean	0..1	attr	This indicates whether the corresponding dataElement shall be transmitted using end-to-end protection. Caveat: The E2E wrapper approach involves technologies that are not subjected to the AUTOSAR standard and is superseded by the superior E2E transformer approach (which is fully standardized by AUTOSAR). Hence, new projects (without legacy constraints due to carry-over parts) shall use the fully standardized E2E transformer approach. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime

Table A.733: ReceiverComSpec

Class	ReceptionComSpecProps			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	This meta-class defines a set of reception attributes which the application software is assumed to implement.			
Base	ARObject			
Aggregated by	ReceiverComSpec.receptionProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dataUpdate Period	TimeValue	0..1	attr	This attribute defines the period in which the application shall check for updated data. This attribute is used for the configuration of the E2E protection, but may also indicate a general data reception period.
timeout	TimeValue	0..1	attr	This attribute defines the time interval after which the application shall assume that the to be received data reception has timed out, i.e. the respective data has not been received for that amount of time.

Table A.734: ReceptionComSpecProps

Class	RecordValueSpecification			
Package	M2::AUTOSARTemplates::CommonStructure::Constants			
Note	Specifies the values for a record.			
Base	ARObject, CompositeValueSpecification, ValueSpecification			
Aggregated by	ApplicationAssocMapElementValueSpecification.key, ApplicationAssocMapElementValueSpecification. value, ArrayValueSpecification.element, CalibrationParameterValue.applInitValue, CalibrationParameter Value.implInitValue, CompositeRuleBasedValueSpecification.argument, ConstantSpecification.value Spec, CryptoServiceKey.developmentValue, DiagnosticEnvDataCondition.compareValue, DiagnosticEnv DataElementCondition.compareValue, FieldSenderComSpec.initValue, ISignal.initValue, ISignal.timeout SubstitutionValue, NonqueuedReceiverComSpec.initValue, NonqueuedReceiverComSpec.timeout SubstitutionValue, NonqueuedSenderComSpec.initValue, NvProvideComSpec.ramBlockInitValue, Nv ProvideComSpec.romBlockInitValue, NvRequireComSpec.initValue, ParameterDataPrototype.initValue, ParameterProvideComSpec.initValue, ParameterRequireComSpec.initValue, PersistencyDataRequired ComSpec.initValue, PersistencyKeyValuePair.initValue, PortDefinedArgumentValue.value, PortPrototype BlueprintInitValue.value, RecordValueSpecification.field, StateManagementCompareCondition.compare Value, SwDataDefProps.invalidValue, VariableDataPrototype.initValue			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	RecordValueSpecification			
field (ordered)	ValueSpecification	*	aggr	The value for a single record field. This could also be mapped explicitly to a record element of the data type using the shortName of the ValueSpecification. But this would introduce a relationship to the data type that is too strong. As of now, it is only important that the structure of the data type matches the structure of the Value Specification independently of the shortNames. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=field, field.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table A.735: RecordValueSpecification

Enumeration	ReentrancyLevelEnum	
Package	M2::AUTOSARTemplates::CommonStructure::InternalBehavior	
Note	Specifies if and in which kinds of environments an entity is reentrant.	
Aggregated by	*ExecutableEntity*.reentrancyLevel	
Literal	**Description**	
multicoreReentrant	Unlimited concurrent execution of this entity is possible, including preemption and parallel execution on multi core systems. **Tags:**atp.EnumerationLiteralIndex=0	
nonReentrant	Concurrent execution of this entity is not possible. **Tags:**atp.EnumerationLiteralIndex=1	
singleCore Reentrant	Pseudo-concurrent execution (i.e. preemption) of this entity is possible on single core systems. **Tags:**atp.EnumerationLiteralIndex=2	

Table A.736: ReentrancyLevelEnum

Primitive	Ref			
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::PrimitiveTypes			

Note	This primitive denotes a name based reference. For detailed syntax see the xsd.pattern.
	
	    • first slash (relative or absolute reference) [optional]
	    • Identifier [required]
	    • a sequence of slashes and Identifiers [optional]
	
	This primitive is used by the meta-model tools to create the references.
	
	Tags:
	xml.xsd.customType=REF
	xml.xsd.pattern=/?[a-zA-Z][a-zA-Z0-9_]{0,127}(/[a-zA-Z][a-zA-Z0-9_]{0,127})*
	xml.xsd.type=string			

Attribute	**Type**	**Mult.**	**Kind**	**Note**
base	Identifier	0..1	attr	This attribute reflects the base to be used for this reference. **Tags:**xml.attribute=true

▽

△

Primitive	Ref			
blueprintValue	String	0..1	attr	This represents a description that documents how the value shall be defined when deriving objects from the blueprint. **Tags:** atp.Status=draft xml.attribute=true
index	PositiveInteger	0..1	attr	This attribute supports the use case to point on specific elements in an array. This is in particular required if arrays are used to implement particular data objects. The counting of array indices starts with the value 0, i.e. the index of the first array element is 0. **Tags:**xml.attribute=true

Table A.737: Ref

Class	ReferenceBase			
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::ARPackage			
Note	This meta-class establishes a basis for relative references. Reference bases are identified by the short Label which shall be unique in the current package.			
Base	ARObject			
Aggregated by	ARPackage.referenceBase			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
globalElement	ReferrableSubtypes Enum	*	attr	This attribute represents a meta-class for which the global referencing is supported via this reference base. **Tags:**xml.sequenceOffset=29
globalIn Package	ARPackage	*	ref	This represents the ability to express that global elements live in various packages which do not have a common ancestor package. Packages mentioned by Reference Base.globalInPackage are used in addition to the one in ReferenceBase.package. **Tags:**xml.sequenceOffset=28
isDefault	Boolean	1	attr	This attribute denotes if the current ReferenceBase is the default. Note that there can only be one default reference base within a package. **Tags:** atp.Status=obsolete xml.sequenceOffset=20
package	ARPackage	0..1	ref	This association specifies the basis of all relative references with the base equals shortLabel. **Tags:**xml.sequenceOffset=30
shortLabel	Identifier	1	attr	This is the name of the reference base. By this name, particular references can denote the applicable base. **Stereotypes:** atpIdentityContributor **Tags:**xml.sequenceOffset=10

Table A.738: ReferenceBase

Class	ReferenceCondition			
Package	M2::AUTOSARTemplates::CommonStructure::StandardizationTemplate::DataExchangePoint::Data FormatTailoring			
Note	The ReferenceCondition evaluates to true, if the referenced reference is accepted by all rules of this condition.			
Base	ARObject, AbstractCondition, AbstractMultiplicityRestriction, AttributeCondition			
Aggregated by	ClassContentConditional.condition, InvertCondition.condition			
Attribute	Type	Mult.	Kind	Note
reference	ReferenceTailoring	1	ref	The reference that has to be accepted by the restrictions of this ReferenceCondition

Table A.739: ReferenceCondition

Class	ReferenceTailoring			
Package	M2::AUTOSARTemplates::CommonStructure::StandardizationTemplate::DataExchangePoint::Data FormatTailoring			
Note	Tailoring of Non-Containment References.			
Base	ARObject, AttributeTailoring, DataFormatElementReference, DataFormatElementScope, Identifiable, MultilanguageReferrable, Referrable, SpecElementReference, SpecElementScope			
Aggregated by	ClassContentConditional.attributeTailoring			
Attribute	Type	Mult.	Kind	Note
typeTailoring	ClassTailoring	*	aggr	Local class tailoring for content that is referenced by this reference.
unresolved Reference Restriction	UnresolvedReference RestrictionWithSeverity	0..1	aggr	Specifies the severity of unresolved references.

Table A.740: ReferenceTailoring

Class	ReferenceValueSpecification			
Package	M2::AUTOSARTemplates::CommonStructure::Constants			
Note	Specifies a reference to a data prototype to be used as an initial value for a pointer in the software.			
Base	ARObject, ValueSpecification			
Aggregated by	ApplicationAssocMapElementValueSpecification.key, ApplicationAssocMapElementValueSpecification. value, ArrayValueSpecification.element, CalibrationParameterValue.applInitValue, CalibrationParameter Value.implInitValue, ConstantSpecification.valueSpec, CryptoServiceKey.developmentValue, Diagnostic EnvDataCondition.compareValue, DiagnosticEnvDataElementCondition.compareValue, FieldSenderCom Spec.initValue, ISignal.initValue, ISignal.timeoutSubstitutionValue, NonqueuedReceiverComSpec.init Value, NonqueuedReceiverComSpec.timeoutSubstitutionValue, NonqueuedSenderComSpec.initValue, NvProvideComSpec.ramBlockInitValue, NvProvideComSpec.romBlockInitValue, NvRequireComSpec.init Value, ParameterDataPrototype.initValue, ParameterProvideComSpec.initValue, ParameterRequireCom Spec.initValue, PersistencyDataRequiredComSpec.initValue, PersistencyKeyValuePair.initValue, Port DefinedArgumentValue.value, PortPrototypeBlueprintInitValue.value, RecordValueSpecification.field, StateManagementCompareCondition.compareValue, SwDataDefProps.invalidValue, VariableData Prototype.initValue			
Attribute	Type	Mult.	Kind	Note
referenceValue	DataPrototype	0..1	ref	The referenced data prototype.

Table A.741: ReferenceValueSpecification

Class	Referrable (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	AtpDefinition, BswDistinguishedPartition, BswModuleCallPoint, BswModuleClientServerEntry, Bsw VariableAccess, CouplingPortTrafficClassAssignment, DiagnosticEnvModeElement, EthernetPriority Regeneration, ExclusiveAreaNestingOrder, HwDescriptionEntity, ImplementationProps, LinSlaveConfig Ident, ModeTransition, MultilanguageReferrable, PncMappingIdent, SingleLanguageReferrable, SoConI PduIdentifier, SocketConnectionBundle, TimeSyncServerConfiguration, TpConnectionIdent			
Attribute	Type	Mult.	Kind	Note
shortName	Identifier	1	attr	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. **Stereotypes:** atpIdentityContributor **Tags:** xml.enforceMinMultiplicity=true xml.sequenceOffset=-100
shortName Fragment	ShortNameFragment	*	aggr	This specifies how the Referrable.shortName is composed of several shortNameFragments. **Tags:**xml.sequenceOffset=-90

Table A.742: Referrable

Class	RoleBasedBswModuleEntryAssignment			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	This class specifies an assignment of a role to a particular BswModuleEntry (usually a configurable callback). With this assignment, the role of the callback is mapped to a specific ServiceNeeds element, so that a tool is able to create appropriate configuration values for the module that implements the AUTOSAR Service.			
Base	ARObject			
Aggregated by	BswServiceDependency.assignedEntryRole			
Attribute	Type	Mult.	Kind	Note
assignedEntry	BswModuleEntry	0..1	ref	The assigned entry. It should be an implementedEntry or expectedEntry of the module or cluster that requires the ServiceNeeds.
role	Identifier	0..1	attr	This is the role of the assigned BswModuleEntry in the given context. The attribute is required (for example) because different kind of callbacks may be associated with the same ServiceNeeds (e.g. end-notification vs. error-notification). The value shall be the role name of a configurable function call (usually a callback) as standardized in the Software Specification of the related AUTOSAR Service.

Table A.743: RoleBasedBswModuleEntryAssignment

Class	RoleBasedDataAssignment			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	This class specifies an assignment of a role to a particular data object in either • the SwcInternalBehavior of a software component (or in the BswInternalBehavior of a BSW module or BSW cluster) in the context of an AUTOSAR Service or • an NvBlockDescriptor to sort out the assignment of event-based writing strategies to data elements in a PortPrototype. With this assignment, the role of the data can be mapped to a DataPrototype that is used in the context of the definition of a specific ServiceNeeds or NvBlockDescriptor, so that a tool is able to create the correct access or writing strategy.			
Base	ARObject			
Aggregated by	BswServiceDependency.assignedData, NvBlockDescriptor.writingStrategy, SwcServiceDependency.assignedData			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
role	Identifier	0..1	attr	This is the role of the assigned data in the given context. Possible values need to be specified on M1 level. Additionally the TPS Software Component Template provides a list of applicable roles for various service dependencies and service use cases in chapter 13 "Service Dependencies and Service Use Cases" (e.g., ramBlock in case of the needs for a permanent RAM block).
usedData Element	AutosarVariableRef	0..1	aggr	The VariableDataPrototype used in this role, e.g. • Permanent RAM Block of an NVRAM Block which shall belong to the same SwcInternal Behavior or BswInternalBehavior. • In the role signalBasedDiagnostics it has to refer to a VariableDataPrototype in a SenderReceiver Interface or a NvDataInterface.
usedParameter Element	AutosarParameterRef	0..1	aggr	The ParameterDataPrototype used in this role, e.g. • ROM Block of an NVRAM Block. It shall belong to the same SwcInternalBehavior or Bsw Internalbehavior. • In the role signalBasedDiagnostics it has to refer to a ParameterDataPrototype in a Parameter Interface.
usedPim	PerInstanceMemory	0..1	ref	The (untyped) PerInstanceMemory used in this role (e.g. as a Permanent RAM Block for an NVRAM Block).

Table A.744: RoleBasedDataAssignment

Class	RoleBasedDataTypeAssignment			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::ServiceMapping			
Note	This class specifies an assignment of a role to a particular data type of a software component (or in the BswModuleBehavior of a module or cluster) in the context of an AUTOSAR Service. With this assignment, the role of the data type can be mapped to a specific ServiceNeeds element, so that a tool is able to create the correct access.			
Base	ARObject			
Aggregated by	ServiceDependency.assignedDataType			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
role	Identifier	0..1	attr	This is the role of the associated data type in the given context.

▽

△

Class	RoleBasedDataTypeAssignment			
used Implementation DataType	ImplementationData Type	0..1	ref	This represents the associated ImplementationDataType.

Table A.745: RoleBasedDataTypeAssignment

Class	RoleBasedPortAssignment			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::ServiceMapping			
Note	This class specifies an assignment of a role to a particular service port (RPortPrototype or PPort Prototype) of an AtomicSwComponentType. With this assignment, the role of the service port can be mapped to a specific ServiceNeeds element, so that a tool is able to create the correct connector.			
Base	ARObject			
Aggregated by	NvBlockDescriptor.clientServerPort, SwcServiceDependency.assignedPort			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
portPrototype	PortPrototype	0..1	ref	Service PortPrototype used in the assigned role. This PortPrototype shall either belong to the same AtomicSw ComponentType as the SwcInternalBehavior which owns the ServiceDependency or to the same NvBlockSw ComponentType as the NvBlockDescriptor.
role	Identifier	0..1	attr	This is the role of the assigned Port in the given context. The value shall be a shortName of the Blueprint of a Port Interface as standardized in the Software Specification of the related AUTOSAR Service.

Table A.746: RoleBasedPortAssignment

Class	RootSwCompositionPrototype			
Package	M2::AUTOSARTemplates::SystemTemplate			
Note	The RootSwCompositionPrototype represents the top-level-composition of software components within a given System. 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. 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. The contained SwComponentPrototypes are fully specified by their SwComponentTypes (including Port Prototypes, PortInterfaces, VariableDataPrototypes, SwcInternalBehavior etc.), and their ports are interconnected using SwConnectorPrototypes.			
Base	ARObject, AtpFeature, AtpPrototype, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	AtpClassifier.atpFeature, System.rootSoftwareComposition			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
calibration ParameterValue Set	CalibrationParameter ValueSet	*	ref	Used CalibrationParameterValueSet for instance specific initialization of calibration parameters. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=calibrationParameterValueSet

▽

△

Class	RootSwCompositionPrototype			
flatMap	FlatMap	0..1	ref	The FlatMap used in the scope of this RootSw CompositionPrototype. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=flatMap
software Composition	CompositionSw ComponentType	1	tref	We assume that there is exactly one top-level composition that includes all Component instances of the system. **Stereotypes:** isOfType

Table A.747: RootSwCompositionPrototype

Class	RoughEstimateHeapUsage			
Package	M2::AUTOSARTemplates::CommonStructure::ResourceConsumption::HeapUsage			
Note	Rough estimation of the heap usage.			
Base	*ARObject*, *HeapUsage*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ResourceConsumption.heapUsage			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
memory Consumption	PositiveInteger	0..1	attr	Rough estimate of the heap usage. Unit: byte.

Table A.748: RoughEstimateHeapUsage

Class	RoughEstimateOfExecutionTime			
Package	M2::AUTOSARTemplates::CommonStructure::ResourceConsumption::ExecutionTime			
Note	Provides a description of a rough estimate on the ExecutionTime.			
Base	*ARObject*, *ExecutionTime*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ResourceConsumption.executionTime			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
additional Information	String	0..1	attr	Provides description on the rough estimate of the ExecutionTime.
estimated ExecutionTime	MultidimensionalTime	0..1	aggr	The estimated execution time.

Table A.749: RoughEstimateOfExecutionTime

Class	RoughEstimateStackUsage			
Package	M2::AUTOSARTemplates::CommonStructure::ResourceConsumption::StackUsage			
Note	Rough estimation of the stack usage.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *StackUsage*			
Aggregated by	ResourceConsumption.stackUsage			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
memory Consumption	PositiveInteger	0..1	attr	Rough estimate of the stack usage. Unit: byte.

Table A.750: RoughEstimateStackUsage

Class	RptComponent			
Package	M2::AUTOSARTemplates::CommonStructure::MeasurementCalibrationSupport::RptSupport			
Note	Description of component instance for which rapid prototyping support is implemented.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	RptSupportData.rptComponent			
Attribute	Type	Mult.	Kind	Note
mcData Assignment	RoleBasedMcData Assignment	*	aggr	Reference to related McDataElement describing the implementation of "RP global buffer", "RP global measurement buffer", "RP enabler flag" and the "RP runnable disabler flag".
rpImplPolicy	RptImplPolicy	0..1	aggr	Describes the implemented code preparation for rapid prototyping at data accesses.
rptExecutable Entity	RptExecutableEntity	*	aggr	ExecutableEntity instance which can be bypassed. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=rptExecutableEntity.shortName, rpt ExecutableEntity.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table A.751: RptComponent

Class	RptContainer			
Package	M2::AUTOSARTemplates::SWComponentTemplate::RPTScenario			
Note	This meta-class defines a byPassPoint and the relation to a rptHook. Additionally it may contain further rptContainers if the byPassPoint is not atomic. For example a byPass Point referencing to a RunnableEntity may contain rptContainers referring to the data access points of the RunnableEntity. The RptContainer structure on M1 shall follow the M1 structure of the Software Component Descriptions. The category attribute denotes which level of the Software Component Description is annotated.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	RapidPrototypingScenario.rptContainer, RptContainer.rptContainer			
Attribute	Type	Mult.	Kind	Note
byPassPoint	AtpFeature	*	iref	byPassPoint describes the required preparation of the host ECU. At a byPassPoint the host ECU shall be capable to communicate with a RPT System in order to support the execution of the rapid prototyping algorithms with the original data calculated by the host system and to replace dedicated results of the host system by the results of the rapid prototyping algorithm. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=byPassPoint.contextElement, byPass Point.target, byPassPoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime **InstanceRef implemented by:**AnyInstanceRef
explicitRpt ProfileSelection	RptProfile	*	ref	This attribute defines the applicable RptProfiles for the specific RptContainer. If not any references to a specific RptProfile is defined, all RptProfiles defined in the Rapid PrototypingScenario are applicable. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=explicitRptProfileSelection

▽

△

Class	RptContainer			
rptContainer	RptContainer	*	aggr	Sub-level rptContainer definitions of this specific rapid prototyping scenario. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=rptContainer.shortName, rpt Container.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
rptExecutable EntityProperties	RptExecutableEntity Properties	0..1	aggr	Describes the required code preparation for rapid prototyping at ExecutableEntity invocation.
rptHook	RptHook	0..1	aggr	The rptHook describes the link between a byPassPoint and the rapid prototyping algorithm. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=rptHook, rptHook.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
rptImplPolicy	RptImplPolicy	0..1	aggr	Describes the required code preparation for rapid prototyping at data accesses.
rptSw Prototyping Access	RptSwPrototyping Access	0..1	aggr	Describes the required accessibility of data and modes by the rapid prototyping tooling.

Table A.752: RptContainer

Class	RptExecutableEntity			
Package	M2::AUTOSARTemplates::CommonStructure::MeasurementCalibrationSupport::RptSupport			
Note	This describes a ExecutableEntity instance which can be bypassed.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	RptComponent.rptExecutableEntity			
Attribute	Type	Mult.	Kind	Note
rptExecutable EntityEvent	RptExecutableEntity Event	*	aggr	ExecutableEntity event instance activation the owning Rpt ExecutableEntity. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=rptExecutableEntityEvent.shortName, rpt ExecutableEntityEvent.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
rptRead	RoleBasedMcData Assignment	*	aggr	read access to a variable **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=rptRead, rptRead.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
rptWrite	RoleBasedMcData Assignment	*	aggr	write access to a variable **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=rptWrite, rptWrite.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
symbol	CIdentifier	0..1	attr	The symbol describing this ExecutableEntity's entry point.

Table A.753: RptExecutableEntity

Class	RptExecutableEntityEvent				
Package	M2::AUTOSARTemplates::CommonStructure::MeasurementCalibrationSupport::RptSupport				
Note	This describes an ExecutableEntity event instance which can be bypassed.				
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable				
Aggregated by	RptExecutableEntity.rptExecutableEntityEvent				
Attribute	Type	Mult.	Kind	Note	
execution Context	RptExecutionContext	*	ref	This describes the context in which the event of the executable entity is executed.	
mcData Assignment	RoleBasedMcData Assignment	*	aggr	Reference to related McDataElements describing the implementation of "RP runnable disabler flag" and "stimulation enabler flag" The possible roles of the RoleBasedMcData Assignment.role attribute are: • RpRunnableDisablerFlag"	
rptEventId	PositiveInteger	0..1	attr	RPT event id used for service points call.	
rptExecutable EntityProperties	RptExecutableEntity Properties	0..1	aggr	Describes the implemented code preparation for rapid prototyping at ExecutableEntity invocation.	
rptImplPolicy	RptImplPolicy	0..1	aggr	Describes the RptImplPolicy of a RptExecutableEvent for service based bypassing.	
rptServicePoint Post	RptServicePoint	*	ref	This describes the applicable Post Service Points for a RTEEvent / BswEvent of a bypassed ExecutableEntity.	
rptServicePoint Pre	RptServicePoint	*	ref	This describes the applicable Pre Service Points for a RTEEvent / BswEvent of a bypassed ExecutableEntity.	

Table A.754: RptExecutableEntityEvent

Class	RptExecutableEntityProperties				
Package	M2::AUTOSARTemplates::SWComponentTemplate::RPTScenario				
Note	Describes the code preparation for rapid prototyping at ExecutableEntity invocation.				
Base	ARObject				
Aggregated by	RptContainer.rptExecutableEntityProperties, RptExecutableEntityEvent.rptExecutableEntityProperties				
Attribute	Type	Mult.	Kind	Note	
maxRptEventId	PositiveInteger	0..1	attr	Highest RPT event id usable for RTE generated service points. This attribute is relevant, if dedicated id range shall be applied to the ExecutableEntitys of a software component or specific ExecutableEntitys.	
minRptEventId	PositiveInteger	0..1	attr	Lowest RPT event id usable for RTE generated service points. This attribute is relevant, if dedicated id range shall be applied to the ExecutableEntitys of a software component or specific ExecutableEntitys.	
rptExecution Control	RptExecutionControl Enum	0..1	attr	This attribute specifies the rapid prototyping control of the executable	
rptServicePoint	RptServicePointEnum	0..1	attr	Enables generation of service points by the RTE generator.	

Table A.755: RptExecutableEntityProperties

Class	RptHook
Package	M2::AUTOSARTemplates::SWComponentTemplate::RPTScenario
Note	This meta-class provide the ability to describe a rapid prototyping hook. This can either be described by an other AUTOSAR system with the category RPT_SYSTEM or as a non AUTOSAR software.

▽

△

Class	RptHook			
Base	ARObject			
Aggregated by	RptContainer.rptHook			
Attribute	Type	Mult.	Kind	Note
codeLabel	CIdentifier	0..1	attr	This attribute provides a code label which is used in the implementation of the hook. For example this can be an C function name or the name of data definition.
mcdIdentifier	NameToken	0..1	attr	This attribute provides an identifier which shall be used in a MCD System to display the Rpt Hook.
rptArHook	AtpFeature	0..1	iref	This describes the hook with the means of another AUTOSAR system. **InstanceRef implemented by:**AnyInstanceRef
sdg	Sdg	*	aggr	This property allows to keep special data which is not represented by the standard model. It can be utilized to keep e.g. tool specific data.

Table A.756: RptHook

Class	RptImplPolicy			
Package	M2::AUTOSARTemplates::SWComponentTemplate::RPTScenario			
Note	Describes the code preparation for rapid prototyping at data accesses.			
Base	ARObject			
Aggregated by	McDataInstance.rptImplPolicy, RptComponent.rpImplPolicy, RptContainer.rptImplPolicy, RptExecutable EntityEvent.rptImplPolicy			
Attribute	Type	Mult.	Kind	Note
rptEnablerImpl Type	RptEnablerImplType Enum	0..1	attr	For Level 2 or Level3 this property determines how the RTE implements the additional "RP enabler" flag.
rptPreparation Level	RptPreparationEnum	0..1	attr	Mandates RP preparation level for access to VariableData Prototype within generated RTE implementation.

Table A.757: RptImplPolicy

Class	RptProfile			
Package	M2::AUTOSARTemplates::SWComponentTemplate::RPTScenario			
Note	The RptProfile describes the common properties of a Rapid Prototyping method.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	RapidPrototypingScenario.rptProfile			
Attribute	Type	Mult.	Kind	Note
maxService PointId	PositiveInteger	0..1	attr	Highest service point id useable for RTE generated service points.
minServicePoint Id	PositiveInteger	0..1	attr	Lowest service point id useable for RTE generated service points.
servicePoint SymbolPost	CIdentifier	0..1	attr	Complete symbol of the function implementing the post service point. This symbol is used for post-build hooking purposes.
servicePoint SymbolPre	CIdentifier	0..1	attr	Complete symbol of the function implementing the pre service point. This symbol is used for post-build hooking purposes.

▽

△

Class	RptProfile			
stimEnabler	RptEnablerImplType Enum	0..1	attr	Defines if the service points support the stimulation enabler. If RptProfile.stimEnabler is "none" then no stimulation enabler is passed to the service function. Otherwise the stimulation enabler will be passed as a parameter.

Table A.758: RptProfile

Class	RptServicePoint			
Package	M2::AUTOSARTemplates::CommonStructure::MeasurementCalibrationSupport::RptSupport			
Note	Description of a Service Point implemented for rapid prototyping.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	RptSupportData.rptServicePoint			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
serviceId	PositiveInteger	0..1	attr	Unique ID (Range: 0 ... 65535) representing the service point.
symbol	CIdentifier	0..1	attr	Complete symbol of the function implementing the service point. This symbol is used for post-build hooking purposes.

Table A.759: RptServicePoint

Class	RptSupportData			
Package	M2::AUTOSARTemplates::CommonStructure::MeasurementCalibrationSupport::RptSupport			
Note	Root element for rapid prototyping support data related to one Implementation artifact on an ECU, in particular the RTE. The rapid prototyping support data may reference to elements provided for Mc SupportData.			
Base	*ARObject*			
Aggregated by	McSupportData.rptSupportData			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
execution Context	RptExecutionContext	*	aggr	Defines an environment for the execution of Executable Entites.
rptComponent	RptComponent	*	aggr	Description of components for which rapid prototyping support is implemented. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=rptComponent.shortName, rpt Component.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
rptServicePoint	RptServicePoint	*	aggr	This aggregation represents the collection of service points associated with the enclosing RptSuportData **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=rptServicePoint.shortName, rptService Point.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table A.760: RptSupportData

Class	RptSwPrototypingAccess			
Package	M2::AUTOSARTemplates::CommonStructure::MeasurementCalibrationSupport::RptSupport			
Note	Describes the accessibility of data and modes by the rapid prototyping tooling.			
Base	ARObject			
Aggregated by	McDataInstance.resultingRptSwPrototypingAccess, RptContainer.rptSwPrototypingAccess			
Attribute	Type	Mult.	Kind	Note
rptHookAccess	RptAccessEnum	0..1	attr	The related data element can be modified using a post-build hooking tool. An ENABLED VariableData Prototype is implicitly READABLE/WRITABLE.
rptReadAccess	RptAccessEnum	0..1	attr	The related data element can be used as input for bypass functionality by RP tool. If rptImplPolicy is not specified then RTE generation shall ensure at least suitable MC read points are created.
rptWriteAccess	RptAccessEnum	0..1	attr	The related data element can be used as output for bypass functionality by RP tool. The data element shall be prepared to rptLevel2 and related write service points are present.

Table A.761: RptSwPrototypingAccess

Enumeration	RteApiReturnValueProvisionEnum
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::AccessCount
Note	This meta-class provides values to control how return values from RTE APIs are provided.
Aggregated by	AbstractAccessPoint.returnValueProvision
Literal	Description
noReturnValue Provided	The RTE API shall not provide a return value. **Tags:**atp.EnumerationLiteralIndex=1
returnValue Provided	The RTE API shall provide a return value. **Tags:**atp.EnumerationLiteralIndex=0

Table A.762: RteApiReturnValueProvisionEnum

Class	RtePluginProps			
Package	M2::AUTOSARTemplates::CommonStructure::FlatMap			
Note	The properties of a communication graph with respect to the utilization of RTE Implementation Plug-in.			
Base	ARObject			
Aggregated by	FlatInstanceDescriptor.rtePluginProps			
Attribute	Type	Mult.	Kind	Note
associated CrossSwCluster ComRtePlugin	EcucContainerValue	0..1	ref	This associates a communication graph to a specific RTE Implementation Plug-in handling cross Software Cluster communication.
associatedRte Plugin	EcucContainerValue	0..1	ref	This associates a communication graph to a specific RTE Implementation Plug-in handling local Software Cluster communication or communication in a non-cluster ECU.

Table A.763: RtePluginProps

Class	<<atpMixed>> **RuleArguments**			
Package	M2::AUTOSARTemplates::CommonStructure::Constants			
Note	This represents the arguments for a rule-based value specification.			
Base	*ARObject*			
Aggregated by	RuleBasedValueSpecification.arguments			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
v	Numerical	0..1	attr	This represents a numerical value for the RuleBased ValueSpecification.
vf	Numerical	0..1	attr	This represents a numerical value for the RuleBased ValueSpecification which may subject to variability. The latest binding time of the VariationPoint shall be pre CompileTime. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime
vt	VerbatimString	0..1	attr	This represents a textual value for the RuleBasedValue Specification.
vtf	NumericalOrText	0..1	aggr	This aggregation represents the ability to provide a value that is either numerical or text which existence is subject to variability. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=vtf, vtf.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table A.764: RuleArguments

Class	**RuleBasedAxisCont**			
Package	M2::AUTOSARTemplates::CommonStructure::Constants			
Note	This represents the values for the axis of a compound primitive (curve, map). For standard and fix axes, SwAxisCont contains the values of the axis directly. The axis values of SwAxisCont with the category COM_AXIS, RES_AXIS are for display only. For editing and processing, only the values in the related GroupAxis are binding.			
Base	*ARObject*			
Aggregated by	ApplicationRuleBasedValueSpecification.swAxisCont			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
category	CalprmAxisCategory Enum	0..1	attr	This category specifies the particular axis types: • STD_AXIS • COM_AXIS • RES_AXIS (swArraysize necessary) **Tags:**xml.sequenceOffset=20
ruleBased Values	RuleBasedValue Specification	0..1	aggr	This represents the rule based value specification for the axis of a compound primitive (curve, map). **Tags:** xml.roleElement=true xml.roleWrapperElement=false xml.sequenceOffset=80 xml.typeWrapperElement=false
swArraysize	ValueList	0..1	aggr	For multidimensional compound primitives (curve, map ...) it is necessary to know the dimensions.They are specified using swArraySize. **Tags:**xml.sequenceOffset=40

▽

△

Class	RuleBasedAxisCont			
swAxisIndex	AxisIndexType	0..1	attr	This property allows to explicitly assign the axis contents to a particular axis. It is specified by numbers where 1 corresponds to the x-axis. It is also possible to derive the axis association from the sequence of the parent. **Tags:**xml.sequenceOffset=50
unit	Unit	0..1	ref	This represents the physical unit of the provided values. **Tags:**xml.sequenceOffset=30

Table A.765: RuleBasedAxisCont

Class	RuleBasedValueCont			
Package	M2::AUTOSARTemplates::CommonStructure::Constants			
Note	This represents the values of a compound primitive (CURVE, MAP, CUBOID, CUBE_4, CUBE_5, VAL_BLK) or an array.			
Base	ARObject			
Aggregated by	ApplicationRuleBasedValueSpecification.swValueCont			
Attribute	Type	Mult.	Kind	Note
ruleBased Values	RuleBasedValue Specification	0..1	aggr	This represents the rule based value specification for the array or compound primitive (CURVE, MAP, CUBOID, CUBE_4, CUBE_5, VAL_BLK). **Tags:** xml.roleElement=true xml.roleWrapperElement=false xml.sequenceOffset=80 xml.typeWrapperElement=false
swArraysize	ValueList	0..1	aggr	This attribute defines the size of each dimension for compound primitives CURVE, MAP, CUBOID, CUBE_4, CUBE_5, COM_AXIS, RES_AXIS, VAL_BLK. For each dimension one value has to be defined, e.g. one in case of COM_AXIS and two or more in case of MAP. **Tags:**xml.sequenceOffset=40
unit	Unit	0..1	ref	This represents the physical unit of the provided values. **Tags:**xml.sequenceOffset=30

Table A.766: RuleBasedValueCont

Class	RuleBasedValueSpecification			
Package	M2::AUTOSARTemplates::CommonStructure::Constants			
Note	This meta-class is used to support a rule-based initialization approach for data types with an array-nature (ApplicationArrayDataType and ImplementationDataType of category ARRAY) or a compound Application PrimitiveDataType (which also boils down to an array-nature).			
Base	ARObject			
Aggregated by	NumericalRuleBasedValueSpecification.ruleBasedValues, RuleBasedAxisCont.ruleBasedValues, RuleBasedValueCont.ruleBasedValues			
Attribute	Type	Mult.	Kind	Note

▽

△

Class	RuleBasedValueSpecification			
arguments	RuleArguments	0..1	aggr	This represents the arguments for the RuleBasedValue Specification. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=arguments, arguments.variationPoint.short Label vh.latestBindingTime=preCompileTime xml.sequenceOffset=30
maxSizeToFill	Integer	0..1	attr	If a rule is chosen which does not fill until the end, this determines until which size the rule shall fill the values. **Tags:**xml.sequenceOffset=40
rule	Identifier	0..1	attr	This denotes the name of the rule of the RuleBasedValue Specification. The rule determines the calculation specification according which the arguments are used to calculated the values. **Tags:**xml.sequenceOffset=20

Table A.767: RuleBasedValueSpecification

Class	RunnableEntity			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior			
Note	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.			
Base	ARObject, AtpClassifier, AtpFeature, AtpStructureElement, ExecutableEntity, Identifiable, Multilanguage Referrable, Referrable			
Aggregated by	AtpClassifier.atpFeature, SwcInternalBehavior.runnable			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
argument (ordered)	RunnableEntity Argument	*	aggr	This represents the formal definition of a an argument to a RunnableEntity.
asynchronous ServerCall ResultPoint	AsynchronousServer CallResultPoint	*	aggr	The server call result point admits a runnable to fetch the result of an asynchronous server call. 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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=asynchronousServerCallResultPoint.short Name, asynchronousServerCallResultPoint.variation Point.shortLabel vh.latestBindingTime=preCompileTime
canBeInvoked Concurrently	Boolean	0..1	attr	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.

▽

△

Class	**RunnableEntity**			
dataRead Access	VariableAccess	*	aggr	RunnableEntity has implicit read access to dataElement of a sender-receiver PortPrototype or nv data of a nv data PortPrototype.
				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.
				Stereotypes: atpSplitable; atpVariation **Tags:** atp.Splitkey=dataReadAccess.shortName, dataRead Access.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
dataReceive PointBy Argument	VariableAccess	*	aggr	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.
				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.
				Stereotypes: atpSplitable; atpVariation **Tags:** atp.Splitkey=dataReceivePointByArgument.shortName, dataReceivePointByArgument.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
dataReceive PointByValue	VariableAccess	*	aggr	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 the return value. The aggregation of dataReceivePointBy Value 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.
				Stereotypes: atpSplitable; atpVariation **Tags:** atp.Splitkey=dataReceivePointByValue.shortName, data ReceivePointByValue.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
dataSendPoint	VariableAccess	*	aggr	RunnableEntity has explicit write access to dataElement of a sender-receiver PortPrototype or nv data of a nv data PortPrototype.
				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.
				Stereotypes: atpSplitable; atpVariation **Tags:** atp.Splitkey=dataSendPoint.shortName, dataSend Point.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	**RunnableEntity**			
dataWrite Access	VariableAccess	*	aggr	RunnableEntity has implicit write access to dataElement of a sender-receiver PortPrototype or nv data of a nv data PortPrototype. 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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=dataWriteAccess.shortName, dataWrite Access.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
external TriggeringPoint	ExternalTriggeringPoint	*	aggr	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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=externalTriggeringPoint.ident.shortName, externalTriggeringPoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
internal TriggeringPoint	InternalTriggeringPoint	*	aggr	The aggregation of InternalTriggeringPoint is subject to variability with the purpose to support the variant existence of internal triggering points in the implementation. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=internalTriggeringPoint.shortName, internal TriggeringPoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
modeAccess Point	ModeAccessPoint	*	aggr	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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=modeAccessPoint.ident.shortName, mode AccessPoint.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
modeSwitch Point	ModeSwitchPoint	*	aggr	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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=modeSwitchPoint.shortName, modeSwitch Point.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	**RunnableEntity**			
parameter Access	ParameterAccess	*	aggr	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. 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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=parameterAccess.shortName, parameter Access.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
readLocal Variable	VariableAccess	*	aggr	The presence of a readLocalVariable implies that a RunnableEntity needs read access to a VariableData Prototype in the role of implicitInterRunnableVariable or explicitInterRunnableVariable. 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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=readLocalVariable.shortName, readLocal Variable.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
serverCallPoint	ServerCallPoint	*	aggr	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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=serverCallPoint.shortName, serverCall Point.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
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.
writtenLocal Variable	VariableAccess	*	aggr	The presence of a writtenLocalVariable implies that a RunnableEntity needs write access to a VariableData Prototype in the role of implicitInterRunnableVariable or explicitInterRunnableVariable. The aggregation of writtenLocalVariable is subject to variability with the purpose to support the conditional existence of implicitInterRunnableVariable and explicit InterRunnableVariable or the variant existence of written LocalVariable (points) in the implementation. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=writtenLocalVariable.shortName, written LocalVariable.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table A.768: RunnableEntity

Class	RunnableEntityArgument
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RunnableEntity
Note	This meta-class represents the ability to provide specific information regarding the arguments to a RunnableEntity.
Base	ARObject
Aggregated by	RunnableEntity.argument

Attribute	Type	Mult.	Kind	Note
symbol	CIdentifier	0..1	attr	This represents the symbol to be generated into the actual signature on the level of the C programming language.

<div align="center">Table A.769: RunnableEntityArgument</div>

Enumeration	RxAcceptContainedIPduEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication
Note	Defines whether this ContainerIPdu has a fixed set of containedIPdus assigned for reception.
Aggregated by	ContainerIPdu.rxAcceptContainedIPdu

Literal	Description
acceptAll	No fixed set of containedIPdus is defined for reception, any known containedIPdu (based on header Id) shall be expected within this ContainerIPdu. **Tags:**atp.EnumerationLiteralIndex=0
acceptConfigured	A fixed set of containedIPdus is defined for reception. Only these assigned containedIPdus (based on headerId) are expected in this ContainerIPdu. If a not assigned containedIPdu is received within this ContainerIPdu this containedIPdu is discarded. **Tags:**atp.EnumerationLiteralIndex=1

<div align="center">Table A.770: RxAcceptContainedIPduEnum</div>

Class	SOMEIPTransformationDescription
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer
Note	The SOMEIPTransformationDescription is used to specify SOME/IP transformer specific attributes.
Base	ARObject, Describable, TransformationDescription
Aggregated by	TransformationTechnology.transformationDescription

Attribute	Type	Mult.	Kind	Note
alignment	PositiveInteger	1	attr	Defines the padding for alignment purposes that will be added by the SOME/IP transformer after the serialized data of the variable data length data element. The alignment shall be specified in Bits.
byteOrder	ByteOrderEnum	1	attr	Defines which byte order shall be serialized by the SOME/IP transformer
interfaceVersion	PositiveInteger	1	attr	The interface version the SOME/IP transformer shall use.

<div align="center">Table A.771: SOMEIPTransformationDescription</div>

Class	<<atpVariation>> SOMEIPTransformationISignalProps
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer
Note	The class SOMEIPTransformationISignalProps specifies ISignal specific configuration properties for SOME/IP transformer attributes.
Base	ARObject, Describable, TransformationISignalProps
Aggregated by	ISignal.transformationISignalProps, ISignalGroup.transformationISignalProps

<div align="center">▽</div>

△

Class	<<atpVariation>> **SOMEIPTransformationISignalProps**			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
implements LegacyString Serialization	Boolean	0..1	attr	This attribute indicates that Strings in the SOME/IP message shall NOT be serialized according to the SOME/IP specification for Strings.
				If this attribute is set to true, BOM and null-termination shall NOT be added in the serialization for Strings in the payload. If this attribute is set to false (or not set) BOM and null-termination shall be added in the serialization for Strings in the payload according to the SOME/IP specification for Strings.
				NOTE! This attribute is not future safe, and will be removed in an upcoming AUTOSAR release!"
				Tags:atp.Status=obsolete
interfaceVersion	PositiveInteger	0..1	attr	The interface version the SOME/IP transformer shall use.
isDynamic LengthFieldSize	Boolean	0..1	attr	This attribute shall be used to determine the wire type in the context of using the TLV encoding.
messageType	SOMEIPMessageType Enum	0..1	attr	The Message Type which shall be placed into the SOME/IP header.
session HandlingSR	SOMEIPTransformer SessionHandlingEnum	0..1	attr	Defines whether the SOME/IP transformer shall use session handling for Sender/Receiver communication.
sizeOfArray LengthFields	PositiveInteger	0..1	attr	The size of all length fields (in Bytes) of fixed-size arrays or dynamic size arrays in the SOME/IP message. This attribute is valid for all available occurrences of fixed-size arrays or dynamic size arrays in the SOME/IP message.
sizeOfString LengthFields	PositiveInteger	0..1	attr	The size of all length fields (in Bytes) of dynamic length strings in the SOME/IP message. This attribute is valid for all available occurrences of strings in the SOME/IP message.
sizeOfStruct LengthFields	PositiveInteger	0..1	attr	The size of all length fields (in Bytes) of structs in the SOME/IP message. This attribute is valid for all available occurrences of structures in the SOME/IP message. For a more fine granular modeling on the level of Data Prototypes the DataPrototypeTransformationProps shall be used.
sizeOfUnion LengthFields	PositiveInteger	0..1	attr	The size of all length fields (in Bytes) of unions in the SOME/IP message. This attribute is valid for all available occurrences of Unions in the SOME/IP message. For a more fine granular modeling on the level of Data Prototypes the DataPrototypeTransformationProps shall be used.
tlvDataId Definition	*TlvDataIdDefinitionSet*	*	ref	This reference identifies the TlvDataIdDefinitions relevant for the enclosing SOMEIPTransformationISignalProps

Table A.772: SOMEIPTransformationISignalProps

Class	**SOMEIPTransformationProps**			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	The class SOMEIPTransformationProps specifies SOME/IP specific configuration properties.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TransformationProps*			
Aggregated by	TransformationPropsSet.transformationProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	SOMEIPTransformationProps			
alignment	PositiveInteger	0..1	attr	Defines the padding for alignment purposes that will be added by the SOME/IP transformer after the serialized data of the variable data length data element. The alignment shall be specified in Bits.
sizeOfArray LengthField	PositiveInteger	0..1	attr	This attribute describes the size of the length field (in Bytes) that will be put in front of the referenced Array in the SOME/IP message.
sizeOfString LengthField	PositiveInteger	0..1	attr	This attribute describes the size of the length field (in Bytes) that will be put in front of the referenced String in the SOME/IP message.
sizeOfStruct LengthField	PositiveInteger	0..1	attr	This attribute describes the size of the length field (in Bytes) that will be put in front of a Structure in the SOME/IP message.
sizeOfUnion LengthField	PositiveInteger	0..1	attr	This attribute describes the size of the length field (in Bytes) that will be put in front of a Union in the SOME/IP message.

Table A.773: SOMEIPTransformationProps

Class	SdgClass			
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::SpecialDataDef			
Note	An SdgClass specifies the name and structure of the SDG that may be used to store proprietary data in an AUTOSAR model. The SdgClass is similar to an UML stereotype.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *SdgElementWithGid*			
Aggregated by	SdgDef.sdgClass			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
attribute (ordered)	SdgAttribute	*	aggr	Defintion of the structure of the Sdg **Tags:**xml.sequenceOffset=30
caption	Boolean	0..1	attr	Specifies if a caption is required. Note: only Sdgs that have a caption can be referenced **Tags:**xml.sequenceOffset=20
extendsMeta Class	MetaClassName	0..1	attr	The AUTOSAR Meta-Class that may be extended by this SdgClass. **Tags:**xml.sequenceOffset=10
sdgConstraint	TraceableText	*	ref	Semantic constraints that restrict the structure of the special data group. **Tags:**xml.sequenceOffset=40

Table A.774: SdgClass

Class	SdgDef
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::SpecialDataDef
Note	A SdgDef groups several SdgClasses which belong to the same extension. The concept of an SdgDef is similiar to an UML Profile. **Tags:**atp.recommendedPackage=SdgDefs
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*

▽

△

Class	SdgDef			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
sdgClass	SdgClass	*	aggr	The owned sdgClasses which define the structure of the Sdgs **Tags:**xml.namePlural=SDG-CLASSES

Table A.775: SdgDef

Primitive	**SectionInitializationPolicyType**
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::PrimitiveTypes
Note	SectionInitializationPolicyType describes the intended initialization of MemorySections. The following values are standardized in AUTOSAR Methodology: • **INIT**: To be used for (explicitly or not explicitly) initialized variables. • **CLEARED**: To be used for not explicitly initialized variables. • **POWER-ON-CLEARED**: To be used for variables that are not explicitly initialized (cleared) during normal start-up. Instead these are cleared only after power on reset. Please note that the values are defined similar to the representation of enumeration types in the XML schema to ensure backward compatibility. **Tags:** xml.xsd.customType=SECTION-INITIALIZATION-POLICY-TYPE xml.xsd.type=NMTOKEN

Table A.776: SectionInitializationPolicyType

Class	**SectionNamePrefix**			
Package	M2::AUTOSARTemplates::CommonStructure::ResourceConsumption::MemorySectionUsage			
Note	A prefix to be used for generated code artifacts defining a memory section name in the source code of the using module or SWC.			
Base	ARObject, *ImplementationProps*, *Referrable*			
Aggregated by	ResourceConsumption.sectionNamePrefix			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
implementedIn	DependencyOnArtifact	0..1	ref	Optional reference that allows to Indicate the code artifact (header file) containing the preprocessor implementation of memory sections with this prefix. The usage of this link supersedes the usage of a memory mapping header with the default name (derived from the BswModuleDescription's shortName).

Table A.777: SectionNamePrefix

Class	**SecureCommunicationAuthenticationProps**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Authentication properties used to configure SecuredIPdus.			
Base	ARObject, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SecureCommunicationPropsSet.authenticationProps			
Attribute	*Type*	*Mult.*	*Kind*	*Note*

▽

△

Class	SecureCommunicationAuthenticationProps			
authInfoTx Length	PositiveInteger	0..1	attr	This attribute defines the length in bits of the authentication code to be included in the payload of the authenticated Pdu.

Table A.778: SecureCommunicationAuthenticationProps

Class	SecureCommunicationFreshnessProps			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Freshness properties used to configure SecuredIPdus.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SecureCommunicationPropsSet.freshnessProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
freshness CounterSync Attempts	PositiveInteger	0..1	attr	This attribute defines the number of Freshness Counter re-synchronization attempts when a verification failed for a Secured I-PDU. If the value is zero, there will be no additional verification attempt to synchronize with a potentially better fitting Freshness Counter value. This attribute is only applicable if useFreshnessTimestamp is FALSE.
freshness TimestampTime PeriodFactor	PositiveInteger	0..1	attr	This attribute defines a factor that specifies the time period for the Freshness Timestamp. It holds a multiplication factor that specifies the concrete meaning of a Freshness Timestamp increment by one on basis of microseconds.
freshnessValue Length	PositiveInteger	0..1	attr	This attribute defines the complete length in bits of the Freshness Value. As long as the key doesn't change the counter shall not overflow. The length of the counter shall be determined based on the expected life time of the corresponding key and frequency of usage of the counter.
freshnessValue TxLength	PositiveInteger	0..1	attr	This attribute defines the length in bits of the Freshness Value to be included in the payload of the Secured I-PDU. This length is specific to the least significant bits of the complete Freshness Counter. If the attribute is 0 no Freshness Value is included in the Secured I-PDU.
useFreshness Timestamp	Boolean	0..1	attr	This attribute specifies whether the Freshness Value is generated through individual Freshness Counters or by a Timestamps. The value is set to TRUE when Timestamps are used.

Table A.779: SecureCommunicationFreshnessProps

Class	SecureCommunicationProps			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	This meta-class contains configuration settings that are specific for an individual SecuredIPdu.			
Base	*ARObject*			
Aggregated by	SecuredIPdu.secureCommunicationProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
authData Freshness Length	PositiveInteger	0..1	attr	This attribute defines the length in bits of the authentic PDU data that is passed to the SWC that verifies and generates the Freshness.

▽

△

Class	SecureCommunicationProps			
authData FreshnessStart Position	PositiveInteger	0..1	attr	This value determines the start position in bits of the Authentic PDU that shall be passed on to the SWC that verifies and generates the Freshness. The bit counting is done according to TPS_SYST_01068.
authentication BuildAttempts	PositiveInteger	0..1	attr	This attribute specifies the number of authentication build attempts.
authentication Retries	PositiveInteger	1	attr	This attribute defines the additional number of authentication attempts that are to be carried out when the generation of the authentication information failed for a given SecuredIPdu. If zero is set than only one authentication attempt is done.
dataId	PositiveInteger	1	attr	This attribute defines a numerical identifier for the Secured I-PDU.
freshnessValue Id	PositiveInteger	0..1	attr	This attribute defines the Id of the Freshness Value. The Freshness Value might be a normal counter or a time value.
messageLink Length	PositiveInteger	0..1	attr	SecOC links an AuthenticIPdu and CryptographicIPdu together by repeating a specific part (Message Linker) of the AuthenticIPdu in the CryptographicIPdu. This attribute defines the length in bits of the messageLinker.
messageLink Position	PositiveInteger	0..1	attr	SecOC links an AuthenticIPdu and CryptographicIPdu together by repeating a specific part (Message Linker) of the AuthenticIPdu in the CryptographicIPdu. This attribute defines the startPosition in bits of the messageLinker.
secondary FreshnessValue Id	PositiveInteger	0..1	attr	This attribute defines the Id of the Secondary Freshness Value. The Secondary Freshness Value might be a normal counter or a time value. Please note that this attribute is for documentation only to allow the configuration of required freshness value manager and no upstream mapping is defined for it.
securedArea Length	PositiveInteger	0..1	attr	This attribute defines the length in bytes of the area within the payload Pdu which will be secured.
securedArea Offset	PositiveInteger	0..1	attr	This attribute defines the start position (offset in byte) of the area within the payload Pdu which will be secured.

Table A.780: SecureCommunicationProps

Class	SecuredIPdu			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	If useAsCryptographicPdu is not set or set to false this IPdu contains the payload of an Authentic IPdu supplemented by additional Authentication Information (Freshness Counter and an Authenticator). If useAsCryptographicPdu is set to true this IPdu contains the Authenticator for a payload that is transported in a separate message. The separate Authentic IPdu is described by the Pdu that is referenced with the payload reference from this SecuredIPdu. **Tags:**atp.recommendedPackage=Pdus			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *IPdu*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Pdu*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
authentication Props	SecureCommunication AuthenticationProps	0..1	ref	Reference to authentication properties that are valid for this SecuredIPdu.

▽

△

Class	SecuredIPdu			
dynamic RuntimeLength Handling	Boolean	0..1	attr	Defines whether the length information for handling this SecuredIPdu with SecuredIPdu.useSecuredPdu Header=noHeader is taken from the configuration or from the actually provided length information during runtime. true: SecuredIPdu length information is taken from the actually provided length information during runtime. false: SecuredIPdu length information is taken from the configuration.
freshnessProps	SecureCommunication FreshnessProps	0..1	ref	Reference to freshness properties that are valid for this SecuredIPdu.
payload	PduTriggering	1	ref	Reference to a Pdu that will be protected against unauthorized manipulation and replay attacks.
secure Communication Props	SecureCommunication Props	1	aggr	Specific configuration properties for this SecuredIPdu.
useAs Cryptographic IPdu	Boolean	0..1	attr	If this attribute is set to true the SecuredIPdu contains the Authentication Information for an AuthenticIPdu that is transmitted in a separate message. The AuthenticIPdu contains the original payload, i.e. the secured data. If this attribute is set to false this SecuredIPdu contains the payload of an Authentic IPdu supplemented by additional Authentication Information.
useSecuredPdu Header	SecuredPduHeader Enum	0..1	attr	This attribute defines the size of the header which is inserted into the SecuredIPdu. If this attribute is set to anything but noHeader, the SecuredIPdu contains the Secured I-PDU Header to indicate the length of the AuthenticIPdu. The AuthenticIPdu contains the original payload, i.e. the secured data.

Table A.781: SecuredIPdu

Class	SecurityEventContextProps			
Package	M2::AUTOSARTemplates::SecurityExtractTemplate			
Note	This meta-class specifies the SecurityEventDefinition to be mapped to an IdsmInstance and adds mapping-dependent properties of this security event valid only for this specific mapping. **Tags:**atp.Status=candidate			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	SecurityEventContextMapping.mappedSecurityEvent			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
contextData	SecurityEventContext Data	0..1	aggr	This aggregation represents the definition of optional context data for security events. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=contextData, contextData.variation Point.shortLabel atp.Status=candidate vh.latestBindingTime=systemDesignTime
default ReportingMode	SecurityEventReporting ModeEnum	0..1	attr	This attribute defines the default reporting mode for the referenced security event. **Tags:**atp.Status=candidate

▽

△

Class	**SecurityEventContextProps**			
persistent Storage	Boolean	0..1	attr	This attribute controls whether qualified reportings of the referenced security event shall be stored persistently by the mapped IdsmInstance or not. **Tags:**atp.Status=candidate
securityEvent	SecurityEventDefinition	0..1	ref	This reference defines the security event that is mapped and enriched by SecurityEventMappingProps with mapping dependent properties. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=securityEvent.securityEventDefinition, securityEvent.variationPoint.shortLabel atp.Status=candidate vh.latestBindingTime=systemDesignTime
sensorInstance Id	PositiveInteger	0..1	attr	This attribute defines the ID of the security sensor that detects the referenced security event. **Tags:**atp.Status=candidate
severity	PositiveInteger	0..1	attr	This attribute defines how critical/severe the referenced security event is. Please note that currently, the severity level meanings of specific integer values is not specified by AUTOSAR but left to the party responsible for the IDS system design (e.g. the OEM). **Tags:**atp.Status=candidate

Table A.782: SecurityEventContextProps

Class	**SegmentPosition**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	The StaticPart and the DynamicPart can be separated in multiple segments within the multiplexed PDU. The ISignalIPdus are copied bit by bit into the MultiplexedIPdu. If the space of the first segment is 5 bits large than the first 5 bits of the ISignalIPdu are copied into this first segment and so on.			
Base	ARObject			
Aggregated by	*MultiplexedPart*.segmentPosition			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
segmentByte Order	ByteOrderEnum	1	attr	This attribute defines the order of the bytes of the segment and the packing into the MultiplexedIPdu. Please consider that [constr_3247] and [constr_3224] are restricting the usage of this attribute.
segmentLength	Integer	1	attr	Data Length of the segment in bits.
segment Position	Integer	1	attr	Segments bit position relatively to the beginning of a multiplexed IPdu. Note that the absolute position of the segment in the MultiplexedIPdu is determined by the definition of the segmentByteOrder attribute of the SegmentPosition. If Big Endian is specified, the start position indicates the bit position of the most significant bit in the IPdu. If Little Endian is specified, the start position indicates the bit position of the least significant bit in the IPdu. In AUTOSAR the bit counting is always set to "sawtooth" and the bit order is set to "Decreasing". The bit counting in byte 0 starts with bit 0 (least significant bit). The most significant bit in byte 0 is bit 7.

Table A.783: SegmentPosition

Class	SenderComSpec (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	Communication attributes for a sender port (PPortPrototype typed by SenderReceiverInterface).			
Base	ARObject, PPortComSpec			
Subclasses	NonqueuedSenderComSpec, QueuedSenderComSpec			
Aggregated by	AbstractProvidedPortPrototype.providedComSpec, PortPrototypeBlueprint.providedComSpec			
Attribute	Type	Mult.	Kind	Note
composite Network Representation	CompositeNetwork Representation	*	aggr	This represents a CompositeNetworkRepresentation defined in the context of a SenderComSpec.
dataElement	AutosarDataPrototype	0..1	ref	Data element these quality of service attributes apply to.
handleOutOf Range	HandleOutOfRange Enum	0..1	attr	This attribute controls how out-of-range values shall be dealt with.
network Representation	SwDataDefProps	0..1	aggr	A networkRepresentation is used to define how the data Element is mapped to a communication bus. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=networkRepresentation
transmission Acknowledge	Transmission Acknowledgement Request	0..1	aggr	Requested transmission acknowledgement for data element.
transmission Props	TransmissionComSpec Props	0..1	aggr	This aggregation represents the definition transmission props in the context of the enclosing SenderComSpec.
usesEndToEnd Protection	Boolean	0..1	attr	This indicates whether the corresponding dataElement shall be transmitted using end-to-end protection. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime

Table A.784: SenderComSpec

Class	SenderRecArrayElementMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	The SenderRecArrayElement may be a primitive one or a composite one. If the element is primitive, it will be mapped to the SystemSignal (multiplicity 1). If the VariableDataPrototype that is referenced by Sender ReceiverToSignalGroupMapping is typed by an ApplicationDataType the reference to the Application ArrayElement shall be used. If the VariableDataPrototype is typed by the ImplementationDataType the reference to the ImplementationArrayElement shall be used. If the element is composite, there will be no mapping to the SystemSignal (multiplicity 0). In this case the ArrayElementMapping element will aggregate the TypeMapping element. In that way also the composite datatypes can be mapped to SystemSignals. Regardless whether composite or primitive array element is mapped the indexed element always needs to be specified.			
Base	ARObject			
Aggregated by	SenderRecArrayTypeMapping.arrayElementMapping			
Attribute	Type	Mult.	Kind	Note
complexType Mapping	SenderRecComposite TypeMapping	0..1	aggr	This aggregation will be used if the element is composite.
indexedArray Element	IndexedArrayElement	1	aggr	Reference to an indexed array element in the context of the dataElement or in the context of a composite element.
systemSignal	SystemSignal	0..1	ref	Reference to the system signal used to carry the primitive ApplicationArrayElement.

Table A.785: SenderRecArrayElementMapping

Class	SenderRecCompositeTypeMapping (abstract)
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping
Note	Two mappings exist for the composite data types: "ArrayTypeMapping" and "RecordTypeMapping". In both, a primitive datatype will be mapped to a system signal. But it is also possible to combine the arrays and the records, so that an "array" could be an element of a "record" and in the same manner a "record" could be an element of an "array". Nesting these data types is also possible. If an element of a composite data type is again a composite one, the "CompositeTypeMapping" element will be used one more time (aggregation between the ArrayElementMapping and CompositeType Mapping or aggregation between the RecordElementMapping and CompositeTypeMapping).
Base	ARObject
Subclasses	SenderRecArrayTypeMapping, SenderRecRecordTypeMapping
Aggregated by	SenderRecArrayElementMapping.complexTypeMapping, SenderReceiverCompositeElementToSignal Mapping.typeMapping, SenderReceiverToSignalGroupMapping.typeMapping, SenderRecRecord ElementMapping.complexTypeMapping

Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.786: SenderRecCompositeTypeMapping

Class	SenderRecRecordElementMapping
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping
Note	Mapping of a primitive record element to a SystemSignal. If the VariableDataPrototype that is referenced by SenderReceiverToSignalGroupMapping is typed by an ApplicationDataType the reference application RecordElement shall be used. If the VariableDataPrototype is typed by the ImplementationDataType the reference implementationRecordElement shall be used. Either the implementationRecordElement or applicationRecordElement reference shall be used. If the element is composite, there will be no mapping to the SystemSignal (multiplicity 0). In this case the RecordElementMapping element will aggregate the complexTypeMapping element. In that way also the composite datatypes can be mapped to SystemSignals.
Base	ARObject
Aggregated by	SenderRecRecordTypeMapping.recordElementMapping

Attribute	Type	Mult.	Kind	Note
application RecordElement	ApplicationRecord Element	0..1	ref	Reference to an ApplicationRecordElement in the context of the dataElement or in the context of a composite element.
complexType Mapping	SenderRecComposite TypeMapping	0..1	aggr	This aggregation will be used if the element is composite.
implementation RecordElement	ImplementationData TypeElement	0..1	ref	Reference to an ImplementationRecordElement in the context of the dataElement or in the context of a composite element.
senderToSignal TextTable Mapping	TextTableMapping	0..1	aggr	This mapping allows for the text-table translation between the sending DataPrototype that is defined in the Port Prototype and the physicalProps defined for the System Signal.
signalTo ReceiverText TableMapping	TextTableMapping	0..1	aggr	This mapping allows for the text-table translation between the physicalProps defined for the SystemSignal and a receiving DataPrototype that is defined in the Port Prototype.
systemSignal	SystemSignal	0..1	ref	Reference to the system signal used to carry the primitive ApplicationRecordElement.

Table A.787: SenderRecRecordElementMapping

Class	**SenderReceiverAnnotation** (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::ApplicationAttributes			
Note	Annotation of the data elements in a port that realizes a sender/receiver interface.			
Base	*ARObject*, *GeneralAnnotation*			
Subclasses	ReceiverAnnotation, SenderAnnotation			
Aggregated by	*PortPrototype*.senderReceiverAnnotation			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
computed	Boolean	0..1	attr	Flag whether this data element was not measured directly but instead was calculated from possibly several other measured or calculated values.
dataElement	VariableDataPrototype	0..1	ref	The instance of VariableDataPrototype annotated.
limitKind	DataLimitKindEnum	0..1	attr	This min or max has not to be mismatched with the min- and max for data-value in a compu-method. For example, this annotation shows when the result of the calculation performed in a RunnableEntity owned by one AtomicSw ComponentType is transmitted to another AtomicSw ComponentType whose RunnableEntity will use this value as a limit, e.g. the max.power which can be used by that software-component, or the current min. slip.
processingKind	ProcessingKindEnum	0..1	attr	This attribute controls how data is processed according to the possible values of ProcessingKindEnum.

Table A.788: SenderReceiverAnnotation

Class	**SenderReceiverCompositeElementToSignalMapping**			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	Mapping of an Variable Data Prototype which is aggregated within a composite datatype to a System Signal (only one element of the composite data type is mapped).			
Base	*ARObject*, *DataMapping*			
Aggregated by	SystemMapping.dataMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dataElement	VariableDataPrototype	0..1	iref	Reference to a data element with a composite datatype from which one element is mapped to a SystemSignal. **InstanceRef implemented by:**VariableDataPrototypeInSystemInstanceRef
systemSignal	SystemSignal	1	ref	Reference to the SystemSignal to which one primitive of the composite type is mapped.
typeMapping	SenderRecComposite TypeMapping	1	aggr	The CompositeTypeMapping maps one VariableData Prototype of the composite data type to a SystemSignal.

Table A.789: SenderReceiverCompositeElementToSignalMapping

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.

$\bigtriangledown$

△

Class	SenderReceiverInterface			
invalidation Policy	InvalidationPolicy	*	aggr	InvalidationPolicy for a particular dataElement
metaDataItem Set	MetaDataItemSet	*	aggr	This aggregation defines fixed sets of meta-data items associated with dataElements of the enclosing Sender ReceiverInterface

Table A.790: SenderReceiverInterface

Class	SenderReceiverToSignalGroupMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	Mapping of a sender receiver communication data element with a composite datatype to a signal group.			
Base	ARObject, DataMapping			
Aggregated by	SystemMapping.dataMapping			
Attribute	Type	Mult.	Kind	Note
dataElement	VariableDataPrototype	1	iref	Reference to a data element with a composite datatype which is mapped to a signal group. **InstanceRef implemented by:** VariableDataPrototypeIn SystemInstanceRef
signalGroup	SystemSignalGroup	1	ref	Reference to the signal group, which contain all primitive datatypes of the composite type
typeMapping	SenderRecComposite TypeMapping	1	aggr	The CompositeTypeMapping maps the ApplicationArray Elements and ApplicationRecordElements to Signals of the SignalGroup.

Table A.791: SenderReceiverToSignalGroupMapping

Class	SenderReceiverToSignalMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	Mapping of a sender receiver communication data element to a signal.			
Base	ARObject, DataMapping			
Aggregated by	SystemMapping.dataMapping			
Attribute	Type	Mult.	Kind	Note
dataElement	VariableDataPrototype	1	iref	Reference to the data element. **InstanceRef implemented by:** VariableDataPrototypeIn SystemInstanceRef
senderToSignal TextTable Mapping	TextTableMapping	0..1	aggr	This mapping allows for the text-table translation between the sending DataPrototype that is defined in the Port Prototype and the physicalProps defined for the System Signal.
signalTo ReceiverText TableMapping	TextTableMapping	0..1	aggr	This mapping allows for the text-table translation between the physicalProps defined for the SystemSignal and a receiving DataPrototype that is defined in the Port Prototype.
systemSignal	SystemSignal	1	ref	Reference to the system signal used to carry the data element.

Table A.792: SenderReceiverToSignalMapping

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*, *Atp Type*, CollectableElement, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *Sw ComponentType*			
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 A.793: SensorActuatorSwComponentType

Enumeration	ServerArgumentImplPolicyEnum
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface
Note	This defines how the argument type of the servers RunnableEntity is implemented.
Aggregated by	ArgumentDataPrototype.serverArgumentImplPolicy
Literal	Description
useArgumentType	The argument type of the RunnableEntity is derived from the AutosarDataType of the Argument Prototype. **Tags:**atp.EnumerationLiteralIndex=0
useVoid	The argument type of the RunnableEntity is void. **Tags:**atp.EnumerationLiteralIndex=2

Table A.794: ServerArgumentImplPolicyEnum

Class	ServerCallPoint (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::ServerCall			
Note	If a RunnableEntity owns a ServerCallPoint it is entitled to invoke a particular ClientServerOperation of a specific RPortPrototype of the corresponding AtomicSwComponentType			
Base	ARObject, *AbstractAccessPoint*, *AtpClassifier*, AtpFeature, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	AsynchronousServerCallPoint, SynchronousServerCallPoint			
Aggregated by	*AtpClassifier*.atpFeature, RunnableEntity.serverCallPoint			
Attribute	Type	Mult.	Kind	Note
operation	ClientServerOperation	0..1	iref	The operation that is called by this runnable. **InstanceRef implemented by:**ROperationInAtomicSwc InstanceRef
timeout	TimeValue	0..1	attr	Time in seconds before the server call times out and returns with an error message. It depends on the call type (synchronous or asynchronous) how this is reported.

Table A.795: ServerCallPoint

Class	ServerComSpec			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	Communication attributes for a server port (PPortPrototype and ClientServerInterface).			
Base	ARObject, PPortComSpec			
Aggregated by	AbstractProvidedPortPrototype.providedComSpec, PortPrototypeBlueprint.providedComSpec			
Attribute	Type	Mult.	Kind	Note
operation	ClientServerOperation	0..1	ref	Operation these communication attributes apply to.
queueLength	PositiveInteger	0..1	attr	Length of call queue on the server side. The queue is implemented by the RTE. The value shall be greater or equal to 1. Setting the value of queueLength to 1 implies that incoming requests are rejected while another request that arrived earlier is being processed.
transformation ComSpecProps	TransformationCom SpecProps	*	aggr	This references the TransformationComSpecProps which define port-specific configuration for data transformation.

Table A.796: ServerComSpec

Class	ServiceDependency (abstract)			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	Collects all dependencies of a software module or component on an AUTOSAR Service related to a specific item (e.g. an NVRAM Block, a diagnostic event etc.). It defines the quality of service (Service Needs) of this item as well as (optionally) references to additional elements. This information is required for tools in order to generate the related basic software configuration and ServiceSwComponentTypes.			
Base	ARObject			
Subclasses	BswServiceDependency, SwcServiceDependency			
Attribute	Type	Mult.	Kind	Note
assignedData Type	RoleBasedDataType Assignment	0..1	aggr	This is the role of the assignment data type in the given context. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=assignedDataType, assignedData Type.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
diagnostic Relevance	ServiceDiagnostic RelevanceEnum	0..1	attr	If this attribute indicates a relevance for diagnostics then the integrator has a much easier time identifying the candidates for the configuration of the diagnostic stack. Example: identification of mode conditions (e.g. communication between application and BswM) relevant for the Dcm.
symbolicName Props	SymbolicNameProps	0..1	aggr	This attribute can be taken to contribute to the creation of symbolic name values.

Table A.797: ServiceDependency

Class	ServiceNeeds (abstract)	
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds	
Note	This expresses the abstract needs that a Software Component or Basic Software Module has on the configuration of an AUTOSAR Service to which it will be connected. "Abstract needs" means that the model abstracts from the Configuration Parameters of the underlying Basic Software.	
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable	

▽

△

Class	ServiceNeeds (abstract)			
Subclasses	BswMgrNeeds, ComMgrUserNeeds, CryptoKeyManagementNeeds, CryptoServiceJobNeeds, Crypto ServiceNeeds, *DiagnosticCapabilityElement*, DltUserNeeds, *DoIpServiceNeeds*, EcuStateMgrUser Needs, ErrorTracerNeeds, FunctionInhibitionAvailabilityNeeds, FunctionInhibitionNeeds, Global SupervisionNeeds, HardwareTestNeeds, IdsMgrCustomTimestampNeeds, IdsMgrNeeds, IndicatorStatus Needs, J1939DcmDm19Support, J1939RmIncomingRequestServiceNeeds, J1939RmOutgoingRequest ServiceNeeds, NvBlockNeeds, SecureOnBoardCommunicationNeeds, SupervisedEntityCheckpoint Needs, SupervisedEntityNeeds, SyncTimeBaseMgrUserNeeds, V2xDataManagerNeeds, V2xFacUser Needs, V2xMUserNeeds, VendorSpecificServiceNeeds			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.798: ServiceNeeds

Class	ServiceProxySwComponentType			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components			
Note	This class provides the ability to express a software-component which provides access to an internal service for remote ECUs. It acts as a proxy for the service providing access to the service. An important use case is the request of vehicle mode switches: Such requests can be communicated via sender-receiver interfaces across ECU boundaries, but the mode manager being responsible to perform the mode switches is an AUTOSAR Service which is located in the Basic Software and is not visible in the VFB view. To handle this situation, a ServiceProxySwComponentType will act as proxy for the mode manager. It will have R-Ports to be connected with the mode requestors on VFB level and Service-Ports to be connected with the local mode manager at ECU integration time. Apart from the semantics, a ServiceProxySwComponentType has these specific properties: • A prototype of it can be mapped to more than one ECUs in the system description. • Exactly one additional instance of it will be created in the ECU-Extract per ECU to which the prototype has been mapped. • For remote communication, it can have only R-Ports with sender-receiver interfaces and 1:n semantics. • There shall be no connectors between two prototypes of any ServiceProxySwComponentType. **Tags:**atp.recommendedPackage=SwComponentTypes			
Base	*ARElement*, ARObject, *AtomicSwComponentType*, *AtpBlueprint*, AtpBlueprintable, *AtpClassifier*, *Atp Type*, CollectableElement, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *Sw ComponentType*			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.799: ServiceProxySwComponentType

Class	ServiceSwComponentType			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components			
Note	ServiceSwComponentType is used for configuring services for a given ECU. Instances of this class are only to be created in ECU Configuration phase for the specific purpose of the service configuration. **Tags:**atp.recommendedPackage=SwComponentTypes			
Base	*ARElement*, ARObject, *AtomicSwComponentType*, *AtpBlueprint*, AtpBlueprintable, *AtpClassifier*, *Atp Type*, CollectableElement, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *Sw ComponentType*			
Aggregated by	ARPackage.element			

▽

△

Class	ServiceSwComponentType			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.800: ServiceSwComponentType

Enumeration	ServiceVersionAcceptanceKindEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances
Note	Defined the possible acceptance kinds for required service instances.
Aggregated by	ConsumedServiceInstance.versionDrivenFindBehavior, RequiredSomeipServiceInstance.versionDrivenFindBehavior
Literal	Description
exactOrAnyMinorVersion	Search for ANY or specific minor version service instance and select either ALL returned service instances (in case of ANY) or exactly the specific minor version service instances defined in required MinorVersion. **Tags:**atp.EnumerationLiteralIndex=0
minimumMinorVersion	Search for ANY minor version service instance and select only those service instances which have an equal or greater minor version than given in requiredMinorVersion. **Tags:**atp.EnumerationLiteralIndex=1

Table A.801: ServiceVersionAcceptanceKindEnum

Enumeration	SignalServiceTranslationControlEnum
Package	M2::AUTOSARTemplates::CommonStructure::SignalServiceTranslation
Note	This enumeration allows to define how the service instance offer/subscribe control shall behave.
Aggregated by	SignalServiceTranslationProps.serviceControl
Literal	Description
allPartialNetworksActive	Defines the start of service control when all specified partial networks are active. **Tags:**atp.EnumerationLiteralIndex=3
anyPartialNetworkActive	Defines the start of service control when any specified partial network is active. **Tags:**atp.EnumerationLiteralIndex=4
partialNetwork	Defines the start of service control when specific partial networks are active. **Tags:** atp.EnumerationLiteralIndex=1 atp.Status=obsolete
serviceDiscovery	Defines the start of service control when other service is available. **Tags:**atp.EnumerationLiteralIndex=2
translationStart	Defines the start of service control at translation start. **Tags:**atp.EnumerationLiteralIndex=0

Table A.802: SignalServiceTranslationControlEnum

Class	SignalServiceTranslationElementProps
Package	M2::AUTOSARTemplates::CommonStructure::SignalServiceTranslation
Note	Defined translation properties for individual mapped elements.
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable
Aggregated by	SignalServiceTranslationEventProps.elementProps

▽

△

Class	SignalServiceTranslationElementProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
element	DataPrototype Reference	0..1	aggr	Reference to the leaf element the SignalService TranslationElementProps apply to.
filter	DataFilter	0..1	aggr	Defines an optional filter to be applied during translation.
transmission Trigger	Boolean	0..1	attr	Defines whether the source element (which is mapped to the referenced element) triggers the sending of the respective payload.

Table A.803: SignalServiceTranslationElementProps

Class	SignalServiceTranslationEventProps			
Package	M2::AUTOSARTemplates::CommonStructure::SignalServiceTranslation			
Note	This element allows to define the properties which are applicable for the signal/service translation event.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SignalServiceTranslationProps.signalServiceTranslationEventProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
elementProps	SignalService TranslationElement Props	*	aggr	Defines properties for a single translated element.
safeTranslation	Boolean	0..1	attr	Defined whether the translation shall happen in a safe way.
secure Translation	Boolean	0..1	attr	Defined whether the translation shall happen in a secure way.
translation Target	VariableDataPrototype	0..1	iref	Reference to a VariableDataPrototype representing the target of signal/service translation. **InstanceRef implemented by:**VariableDataPrototypeIn SystemInstanceRef

Table A.804: SignalServiceTranslationEventProps

Class	SignalServiceTranslationProps			
Package	M2::AUTOSARTemplates::CommonStructure::SignalServiceTranslation			
Note	This element allows to define the properties which are applicable for the signal/service translation service.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SignalServiceTranslationPropsSet.signalServiceTranslationProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
control Consumed EventGroup	ConsumedEventGroup	*	ref	Reference to the EventGroup which encapsulates the signal-based payload.
controlPnc	PncMappingIdent	*	ref	Reference to the PNCs which control the offer/subscribe behavior of the translated service instance.
controlProvided EventGroup	EventHandler	*	ref	Reference to the provided event group (aka Event Handler) which is automatically available when service Control equals translationStart.
serviceControl	SignalService TranslationControlEnum	0..1	attr	Defines how the service instance control shall behave.

▽

△

Class	SignalServiceTranslationProps			
signalService Translation EventProps	SignalService TranslationEventProps	*	aggr	Defines properties for a single translated event.

Table A.805: SignalServiceTranslationProps

Class	SimulatedExecutionTime			
Package	M2::AUTOSARTemplates::CommonStructure::ResourceConsumption::ExecutionTime			
Note	Specifies the ExecutionTime which has been gathered using simulation means.			
Base	*ARObject*, *ExecutionTime*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ResourceConsumption.executionTime			
Attribute	Type	Mult.	Kind	Note
maximum ExecutionTime	MultidimensionalTime	0..1	aggr	The maximum simulated execution time.
minimum ExecutionTime	MultidimensionalTime	0..1	aggr	The minimum simulated execution time.
nominal ExecutionTime	MultidimensionalTime	0..1	aggr	The nominal simulated execution time.

Table A.806: SimulatedExecutionTime

Class	SoAdConfig			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances			
Note	SoAd Configuration for one specific Physical Channel.			
Base	*ARObject*			
Aggregated by	EthernetPhysicalChannel.soAdConfig			
Attribute	Type	Mult.	Kind	Note
connection	SocketConnection	*	aggr	This aggregation is obsolete and will be removed in the future. The connectionGroup aggregation with bundled Connections shall be used instead. Old description: Collection of socket connections. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=connection, connection.variationPoint.short Label atp.Status=obsolete vh.latestBindingTime=postBuild
connection Bundle	SocketConnection Bundle	*	aggr	Collection of SocketConnectionBundles. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=connectionBundle.shortName, connection Bundle.variationPoint.shortLabel atp.Status=obsolete vh.latestBindingTime=postBuild
socketAddress	SocketAddress	1..*	aggr	Collection of SoAdAddresses. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=socketAddress.shortName, socket Address.variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.807: SoAdConfig

Class	SoConIPduIdentifier			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances			
Note	Identification of Pdu content on a socket connection. This Identifier is required in case that multiple Pdus are transmitted over the same socket connection.			
Base	ARObject, *Referrable*			
Aggregated by	SocketConnectionIpduIdentifierSet.iPduIdentifier			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
headerId	PositiveInteger	0..1	attr	If multiple Pdus are transmitted over the same connection this headerId can be used to distinguish between the different Pdus. For the constraints on constructing the headerId for SOME/IP also see PRS_SOMEIP_00245.
pduCollection PduTimeout	TimeValue	0..1	attr	Defines the timeout in seconds the PDU collection shall be transmitted at the latest after this PDU has been put into the buffer.
pduCollection Semantics	PduCollection SemanticsEnum	0..1	attr	Specifies if the referenced PduTriggering shall be collected using a queued (i.e. all PDU instances) or last-is-best (i.e. only the last PDU instance) semantics. If this attribute is not present the behavior of "queued" is assumed.
pduCollection Trigger	PduCollectionTrigger Enum	0..1	attr	Defines whether the referenced Pdu contributes to the triggering of the socket transmission if Pdu collection is enabled for this socket.
pduTriggering	PduTriggering	0..1	ref	Reference to a Pdu that is transmitted over a socket connection.

Table A.808: SoConIPduIdentifier

Class	SocketAddress			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances			
Note	This meta-class represents a socket address towards the rest of the meta-model. The actual semantics of the represented socket address, however, is contributed by aggregation of an ApplicationEndpoint.			
Base	ARObject, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SoAdConfig.socketAddress			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
allowedIPv6Ext Headers	IPv6ExtHeaderFilterList	0..1	ref	Reference to a list of IPv6 Extension Headers allowed for this SocketConnection. If no list is referenced all IPv6 Extension Headers are allowed and processed.
allowedTcp Options	TcpOptionFilterList	0..1	ref	Reference to a list of TCP options allowed for this Socket Connection.
application Endpoint	ApplicationEndpoint	0..1	aggr	Application addressing
connector	EthernetCommunication Connector	0..1	ref	Association to a CommunicationConnector in the topology description. This reference shall be used if the SocketAddress describes an IP unicast address for an ECU that is part of the model.
differentiated ServiceField	PositiveInteger	0..1	attr	The 6-bit Differentiated Service Field in the IP headers may be used for classifying network traffic. If not set a value of zero is used to indicate packets that have not been classified.
flowLabel	PositiveInteger	0..1	attr	The 20-bit Flow Label field in the IPv6 header may be used by a source to label sequences of packets for which it requests special handling by the IPv6 routers, such as non-default quality of service. If not set a Flow Label of zero is used to indicate packets that have not been labeled.

$\bigtriangledown$

△

Class	SocketAddress			
multicast Connector	EthernetCommunication Connector	*	ref	Association to a CommunicationConnector in the topology description. This reference shall be used if the Socket Address describes an IP multicast address. This multicast SocketAddress contains references to those ECUs in the model that want to receive the multicast messages.
pathMtu Discovery Enabled	Boolean	0..1	attr	Defines whether the Path MTU Discovery shall be performed for the related socket.
pduCollection MaxBufferSize	PositiveInteger	0..1	attr	Defines the maximum buffer size in Byte which shall be filled before a socket with Pdu collection enabled shall be transmitted to the lower layer.
pduCollection Timeout	TimeValue	0..1	attr	Defines the time in seconds which shall pass before a socket with Pdu collection enabled shall be transmitted to the lower layer after the first Pdu has been put into the socket buffer.
staticSocket Connection	StaticSocketConnection	*	aggr	Definition of a static SocketConnection. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=staticSocketConnection.shortName, static SocketConnection.variationPoint.shortLabel vh.latestBindingTime=postBuild
udpChecksum Handling	UdpChecksum CalculationEnum	0..1	attr	Specifies if UDP checksum handling shall be enabled (udpChecksumEnabled) or skipped (udpChecksum Disabled) on the related socket connection.

Table A.809: SocketAddress

Class	SocketConnection			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ObsoleteModel			
Note	The SoAd serves as a (De)Multiplexer between different PDU sources and the TCP/IP stack. **Tags:**atp.Status=obsolete			
Base	*ARObject*, *Describable*			
Aggregated by	SoAdConfig.connection, SocketConnectionBundle.bundledConnection			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
clientIpAddr From Connection Request	Boolean	0..1	attr	If set to true the Server "learns" the client IP address on connection request. This means that the statically configured IP Address of the related client shall be ignored. If set to false the Server only accepts statically configured IP address, e.g. 192.168.1.2. This means that the statically configured IP Address of the Client shall be used.
clientPort	SocketAddress	0..1	ref	Client Port for TCP/UDP connection in an abstract communication sense. The client is the major requester of the communication. Please note that the client may also produce data. **Tags:**atp.Status=obsolete
clientPortFrom Connection Request	Boolean	0..1	attr	If set to true the Server "learns" the client Port on connection request. This means that the statically configured Port of the related client shall be ignored. If set to false the Server only accepts statically configured Port. This means that the statically configured Port of the Client shall be used.

▽

△

Class	SocketConnection			
pdu	SocketConnectionIpdu Identifier	*	aggr	PDUs handed over by the PDU Router (Transmission over the Ethernet) or PDUs handed over by SoAd (Reception over Ethernet). Multiple IPdus can be transmitted over one socket connection. **Tags:**atp.Status=obsolete
pduCollection MaxBufferSize	PositiveInteger	0..1	attr	Defines the maximum buffer size in Byte which shall be filled before a socket with Pdu collection enabled shall be transmitted to the lower layer.
pduCollection Timeout	TimeValue	0..1	attr	Defines the time in seconds which shall pass before a socket with Pdu collection enabled shall be transmitted to the lower layer after the first Pdu has been put into the socket buffer.
runtimeIp Address Configuration	RuntimeAddress ConfigurationEnum	0..1	attr	This attribute determines which protocol is used by the client to obtain the IP Address information. If this attribute is not set to none the value determines the service used by the client to obtain the IP Address information for the SocketConnection. If this attribute is set to none the client used the statically configured IP Address information.
runtimePort Configuration	RuntimeAddress ConfigurationEnum	0..1	attr	This attribute determines which protocol is used by the client to obtain the Port information. If this attribute is not set to none the value determines the service used by the client to obtain the Port information for the Socket Connection. If this attribute is set to none the client uses the statically configured Port information.
shortLabel	Identifier	0..1	attr	This attribute specifies an identifying shortName for the SocketConnection. It shall be unique within its context.

Table A.810: SocketConnection

Class	SoftwareContext			
Package	M2::AUTOSARTemplates::CommonStructure::ResourceConsumption			
Note	Specifies the context of the software for this resource consumption.			
Base	ARObject			
Aggregated by	*ExecutionTime*.softwareContext, *HeapUsage*.softwareContext, *StackUsage*.softwareContext			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
input	String	0..1	attr	Specifies the input vector which is used to provide the ExecutionTime.
state	String	0..1	attr	Specifies the state the software is in when the Execution Time is provided.

Table A.811: SoftwareContext

Class	SomeipSdClientEventGroupTimingConfig			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances			
Note	This meta-class is used to specify configuration related to service discovery in the context of an event group on SOME/IP. **Tags:**atp.recommendedPackage=SomeipSdTimingConfigs			
Base	*ARElement*, ARObject, CollectableElement, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	SomeipSdClientEventGroupTimingConfig			
request ResponseDelay	RequestResponseDelay	0..1	aggr	The Service Discovery shall delay answers to unicast messages triggered by multicast messages (e.g. Subscribe Eventgroup after Offer Service).
subscribe Eventgroup RetryDelay	TimeValue	0..1	attr	This attribute defines the interval in seconds to re-trigger a subscription to a Eventgroup, if a retry to subscribe to a Eventgroup is configured (subscribeEventgroupRetryMax > 0).
subscribe Eventgroup RetryMax	PositiveInteger	0..1	attr	This attribute define the maximum counts of retries to subscribe to an Eventgroup. If the value is set to 0 no retry shall be done. If the value is set to 255 the retry shall be done as along as the Eventgroup is requested and no SubscribeEventGroupAck was received.
timeToLive	PositiveInteger	1	attr	Defines the time in seconds the subscription of this event is expected by the client. this value is sent from the client to the server in the SD-subscribeEvent message.

Table A.812: SomeipSdClientEventGroupTimingConfig

Class	SomeipSdServerServiceInstanceConfig			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances			
Note	Server specific settings that are relevant for the configuration of SOME/IP Service-Discovery.			
	Tags:atp.recommendedPackage=SomeipSdTimingConfigs			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
initialOffer Behavior	InitialSdDelayConfig	0..1	aggr	Controls offer behavior of the server.
offerCyclicDelay	TimeValue	0..1	attr	Optional attribute to define cyclic offers. Cyclic offer is active, if the delay is set (in seconds) and greater then 0.
priority	PositiveInteger	0..1	attr	This attribute defines the VLAN frame priority for Service Discovery messages that result from ProvidedSomeip ServiceInstances that are referencing the SomeipSd ServerServiceInstanceConfig (OfferService, StopOffer Service, SubscribeEventGroupAck). Values from 0 (best effort) to 7 (highest) are allowed.
request ResponseDelay	RequestResponseDelay	0..1	aggr	Maximum/Minimum allowable response delay to entries received by multicast in seconds. The Service Discovery shall delay answers to entries that were transported in a multicast SOME/IP-SD message (e.g. FindService).
serviceOffer TimeToLive	PositiveInteger	1	attr	Defines the time in seconds the service offer is valid.

Table A.813: SomeipSdServerServiceInstanceConfig

Class	SomeipServiceVersion
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances
Note	This meta-class represents the ability to describe a version of a SOME/IP Service.
Base	*ARObject*
Aggregated by	ConsumedServiceInstance.blocklistedVersion, RequiredSomeipServiceInstance.blocklistedVersion, SomeipServiceInterfaceDeployment.serviceInterfaceVersion

▽

△

Class	SporadicEventTriggering			
maximumInter ArrivalTime	MultidimensionalTime	0..1	aggr	The maximum time distance between two consecutive occurrences of the associated event. **Tags:**xml.sequenceOffset=20
minimumInter ArrivalTime	MultidimensionalTime	0..1	aggr	The minimum time distance between two consecutive occurrences of the associated event. **Tags:**xml.sequenceOffset=10
period	MultidimensionalTime	0..1	aggr	The period of the event occurrence. **Tags:**xml.sequenceOffset=40

Table A.817: SporadicEventTriggering

Class	StaticPart			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Some parts/signals of the I-PDU may be the same regardless of the selector field. Such a part is called static part. The static part is optional.			
Base	ARObject, MultiplexedPart			
Aggregated by	MultiplexedIPdu.staticPart			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
iPdu	ISignalIPdu	1	ref	Reference to a Com IPdu which is routed to the IPduM module and is combined to a multiplexedPdu.

Table A.818: StaticPart

Class	StaticSocketConnection			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::ServiceInstances			
Note	Definition of static SocketConnection between the Socket that is defined by the aggregating Socket Address and the remoteAddress.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	SocketAddress.staticSocketConnection			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
iPduIdentifier	SoConIPduIdentifier	*	ref	Assignment of IPduIdentifiers that are transmitted over the static SocketConnection. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=iPduIdentifier.soConIPduIdentifier, iPdu Identifier.variationPoint.shortLabel vh.latestBindingTime=postBuild
remoteAddress	SocketAddress	0..1	ref	RemoteAddress of the static SocketConnection. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=remoteAddress.socketAddress, remote Address.variationPoint.shortLabel vh.latestBindingTime=postBuild
tcpConnect Timeout	TimeValue	0..1	attr	Specifies the time in seconds how long TCP connect attempts are repeated to reach SOAD_SOCON_ONLINE. This attribute is restricted to socket connection groups which are initiating a TCP connection and are under control of SoAd.
tcpRole	TcpRoleEnum	0..1	attr	Defines whether the local Address (that is aggregating the StaticSocketConnection) does a listen or a connect.

Table A.819: StaticSocketConnection

Class	Std			
Package	M2::MSR::Documentation::TextModel::InlineTextElements			
Note	This represents a reference to external standards.			
Base	*ARObject*, *Referrable*, *SingleLanguageReferrable*			
Aggregated by	MixedContentForParagraph.std			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
date	DateTime	0..1	attr	This element specifies the release date of the external standard if applicable. **Tags:**xml.sequenceOffset=50
position	String	0..1	attr	This represents the reference to the relevant positions of a standard. Kept as a string. **Tags:**xml.sequenceOffset=70
state	String	0..1	attr	This represents version and state of a standard. Kept as a string. **Tags:**xml.sequenceOffset=40
subtitle	String	0..1	attr	This represents the subtitle of the standard. **Tags:**xml.sequenceOffset=30
url	Url	0..1	aggr	This represents the URL of the standard. **Tags:**xml.sequenceOffset=60

Table A.820: Std

Class	SubElementMapping			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	This meta-class allows for the definition of mappings of elements of a composite data type.			
Base	*ARObject*			
Aggregated by	DataPrototypeMapping.subElementMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
firstElement	SubElementRef	0..1	aggr	This represents the first element referenced in the scope of the mapping. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=firstElement, firstElement.variation Point.shortLabel vh.latestBindingTime=preCompileTime
secondElement	SubElementRef	0..1	aggr	This represents the second element referenced in the scope of the mapping. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=secondElement, secondElement.variation Point.shortLabel vh.latestBindingTime=preCompileTime
textTable Mapping	TextTableMapping	0..2	aggr	This allows for the text-table translation of individual elements of a composite data type.

Table A.821: SubElementMapping

Class	SupervisedEntityNeeds	
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds	
Note	Specifies the abstract needs on the configuration of the Watchdog Manager for one specific Supervised Entity.	

▽

△

Class	SupervisedEntityNeeds			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *ServiceNeeds*			
Aggregated by	BswServiceDependency.serviceNeeds, SwcServiceDependency.serviceNeeds			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
activateAtStart	Boolean	0..1	attr	true/false: supervision activation status of Supervised Entity shall be enabled/disabled at start.
checkpoints	SupervisedEntity CheckpointNeeds	*	ref	This reference indicates the checkpoints belonging to the Supervised Entity. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=checkpoints.supervisedEntityCheckpoint Needs, checkpoints.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
enable Deactivation	Boolean	0..1	attr	true: software-component shall be allowed to deactivate supervision of this SupervisedEntity false: software-component shall be not allowed to deactivate supervision of this SupervisedEntity
expectedAlive Cycle	TimeValue	0..1	attr	Expected cycle time of alive trigger of this Supervised Entity (in seconds).
maxAliveCycle	TimeValue	0..1	attr	Maximum cycle time of alive trigger of this Supervised Entity (in seconds).
minAliveCycle	TimeValue	0..1	attr	Minimum cycle time of alive trigger of this Supervised Entity (in seconds).
toleratedFailed Cycles	PositiveInteger	0..1	attr	Number of consecutive failed alive cycles for this SupervisedEntity which shall be tolerated until the supervision status of the SupervisedEntity is set to WDGM_ALIVE_EXPIRED (see SWS WdgM for more details). Note that this value has to be recalculated with respect to the WdgM's own cycle time for ECU configuration.

Table A.822: SupervisedEntityNeeds

Enumeration	SupportBufferLockingEnum	
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::PortAPIOptions	
Note	This enumeration represents the ability to define the buffer locking behavior.	
Aggregated by	CommunicationBufferLocking.supportBufferLocking	
Literal	**Description**	
doesNotSupport BufferLocking	Buffer locking is not supported. **Tags:**atp.EnumerationLiteralIndex=0	
supportsBuffer Locking	Buffer locking is supported. **Tags:**atp.EnumerationLiteralIndex=1	

Table A.823: SupportBufferLockingEnum

Class	SwAddrMethod
Package	M2::MSR::DataDictionary::AuxillaryObjects
Note	Used to assign a common addressing method, e.g. common memory section, to data or code objects. These objects could actually live in different modules or components. **Tags:**atp.recommendedPackage=SwAddrMethods

▽

△

Class	SwAddrMethod			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, AtpBlueprintable, CollectableElement, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
memory Allocation KeywordPolicy	MemoryAllocation KeywordPolicyType	0..1	attr	Enumeration to specify the name pattern of the Memory Allocation Keyword.
option	Identifier	*	attr	This attribute introduces the ability to specify further intended properties of the MemorySection in with the related objects shall be placed. These properties are handled as to be selected. The intended options are mentioned in the list. In the Memory Mapping configuration, this option list is used to determine an appropriate MemMapAddressing ModeSet.
section Initialization Policy	SectionInitialization PolicyType	0..1	attr	Specifies the expected initialization of the variables (inclusive those which are implementing VariableData Prototypes). Therefore this is an implementation constraint for initialization code of BSW modules (especially RTE) as well as the start-up code which initializes the memory segment to which the AutosarData Prototypes referring to the SwAddrMethod's are later on mapped. If the attribute is not defined it has the identical semantic as the attribute value "INIT"
sectionType	MemorySectionType	0..1	attr	Defines the type of memory sections which can be associated with this addressing method.

Table A.824: SwAddrMethod

Class	SwAxisCont			
Package	M2::MSR::CalibrationData::CalibrationValue			
Note	This represents the values for the axis of a compound primitive (curve, map). For standard and fix axes, SwAxisCont contains the values of the axis directly. The axis values of SwAxisCont with the category COM_AXIS, RES_AXIS are for display only. For editing and processing, only the values in the related GroupAxis are binding.			
Base	*ARObject*			
Aggregated by	ApplicationValueSpecification.swAxisCont			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
category	CalprmAxisCategory Enum	0..1	attr	This category specifies the particular axis types: • STD_AXIS • COM_AXIS • RES_AXIS (swArraysize necessary) **Tags:**xml.sequenceOffset=20
swArraysize	ValueList	0..1	aggr	For multidimensional compound primitivies (curve, map ...) it is necessary to know the dimensions.They are specified using swArraySize. • RES_AXIS **Tags:**xml.sequenceOffset=70

▽

$\triangle$

Class	SwAxisCont				
swAxisIndex	AxisIndexType	0..1	attr	This property allows to explicitly assign the axis contents to a particular axis. It is specified by numbers where 1 corresponds to the x-axis. It is also possible to derive the axis association from the sequence of the parent. **Tags:**xml.sequenceOffset=50	
swValuesPhys	SwValues	0..1	aggr	swValuesPhys represents the values in the physical domain. **Tags:**xml.sequenceOffset=80	
unit	Unit	0..1	ref	This represents the physical unit of the provided values. **Tags:**xml.sequenceOffset=30	
unitDisplay Name	SingleLanguageUnit Names	0..1	aggr	This represents the display name which is used for the physical unit of the axis. **Tags:**xml.sequenceOffset=40	

Table A.825: SwAxisCont

Class	SwAxisGeneric			
Package	M2::MSR::DataDictionary::Axis			
Note	This meta-class defines a generic axis. In a generic axis the axispoints points are calculated in the ECU. The ECU is equipped with a fixed calculation algorithm. Parameters for the algorithm can be stored in the data component of the ECU. Therefore these parameters are specified in the data declaration, not in the calibration data.			
Base	ARObject			
Aggregated by	SwAxisIndividual.swAxisGeneric			
Attribute	Type	Mult.	Kind	Note
swAxisType	SwAxisType	0..1	ref	Associated axis calculation strategy. **Tags:**xml.sequenceOffset=20
swGenericAxis Param	SwGenericAxisParam	*	aggr	Specific parameter of a generic axis. **Tags:** xml.roleElement=true xml.roleWrapperElement=true xml.sequenceOffset=40 xml.typeElement=false xml.typeWrapperElement=false

Table A.826: SwAxisGeneric

Class	SwAxisGrouped			
Package	M2::MSR::DataDictionary::Axis			
Note	An SwAxisGrouped is an axis which is shared between multiple calibration parameters.			
Base	ARObject, SwCalprmAxisTypeProps			
Aggregated by	SwCalprmAxis.swCalprmAxisTypeProps			
Attribute	Type	Mult.	Kind	Note
sharedAxisType	ApplicationPrimitive DataType	0..1	ref	This is the datatype of the calibration parameter providing the shared axis.

∇

△

Class	SwAxisGrouped			
swAxisIndex	AxisIndexType	0..1	attr	Describes which axis of the referenced calibration parameter provides the values for the group axis. The index satisfies the following convention: • 0 = value axis. in this case, the interpolation result of the referenced parameter is used as a base point index. • The index should only be specified if the parameter under swCalprm contains more than one axis. It is standard practice for the axis index of parameters with more than one axis, to be set to 1, if data has not been assigned to swAxis Index. **Tags:**xml.sequenceOffset=20
swCalprmRef	SwCalprmRefProxy	1	aggr	This property specifes the calibration parameter which serves as the input axis. In AUTOSAR, the type of the referenced Calibration parameter shall be compatible to the type specified by sharedAxisType. **Tags:** xml.roleElement=false xml.roleWrapperElement=false xml.sequenceOffset=30 xml.typeElement=false xml.typeWrapperElement=false

Table A.827: SwAxisGrouped

Class	SwAxisIndividual			
Package	M2::MSR::DataDictionary::Axis			
Note	This meta-class describes an axis integrated into a parameter (field etc.). The integration makes this individual to each parameter. The so-called grouped axis represents the counterpart to this. It is conceived as an independent parameter (see class SwAxisGrouped).			
Base	ARObject, SwCalprmAxisTypeProps			
Aggregated by	SwCalprmAxis.swCalprmAxisTypeProps			
Attribute	Type	Mult.	Kind	Note
compuMethod	CompuMethod	0..1	ref	This is the compuMethod which is expected for the axis. It is used in early stages if the particular input-value is not yet available. **Tags:**xml.sequenceOffset=30
dataConstr	DataConstr	0..1	ref	Refers to constraints, e.g. for plausibility checks. **Tags:**xml.sequenceOffset=80
inputVariable Type	ApplicationPrimitive DataType	0..1	ref	This is the datatype of the input value for the axis. This allows to define e.g. a type of curve, where the input value is finalized at the access point.
swAxisGeneric	SwAxisGeneric	0..1	aggr	this specifies the properties of a generic axis if applicable. **Tags:**xml.sequenceOffset=90
swMaxAxis Points	Integer	0..1	attr	Maximum number of base points contained in the axis of a map or curve. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=60

▽

△

Class	SwAxisIndividual			
swMinAxis Points	Integer	0..1	attr	Minimum number of base points contained in the axis of a map or curve. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=70
swVariableRef (ordered)	SwVariableRefProxy	*	aggr	Refers to input variables of the axis. It is possible to specify more than one variable. Here the following is valid: • The variable with the highest priority shall be given first. It is used in the generation of the code and is also displayed first in the application system. • All variables referenced shall be of the same physical nature. This is usually detected in that the conversion formulae affected refer back to the same SI-units. In AUTOSAR this ensured by the constraint, that the referenced input variables shall use a type compatible to "inputVariableType". • This multiple referencing allows a base point distribution for more than one input variable to be used. One example of this are the temperature curves which can depend both on the induction air temperature and the engine temperature. These variables can be displayed simultaneously by MCD systems (adjustment systems), enabling operating points to be shown in the curves. **Tags:** xml.roleElement=false xml.roleWrapperElement=true xml.sequenceOffset=20 xml.typeElement=false xml.typeWrapperElement=false
unit	Unit	0..1	ref	This represents the physical unit of the input value of the axis. It is provided to support the case that the particular input variable is not yet known. **Tags:**xml.sequenceOffset=40

Table A.828: SwAxisIndividual

Class	SwBaseType			
Package	M2::MSR::AsamHdo::BaseTypes			
Note	This meta-class represents a base type used within ECU software. **Tags:**atp.recommendedPackage=BaseTypes			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpBlueprintable*, *BaseType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.829: SwBaseType

Enumeration	SwCalibrationAccessEnum
Package	M2::MSR::DataDictionary::DataDefProperties
Note	Determines the access rights to a data object w.r.t. measurement and calibration.
Aggregated by	ModeDeclarationGroupPrototype.swCalibrationAccess, SwCalprmAxis.swCalibrationAccess, SwDataDefProps.swCalibrationAccess
Literal	Description
notAccessible	The element will not be accessible via MCD tools, i.e. will not appear in the ASAP file. **Tags:**atp.EnumerationLiteralIndex=0
readOnly	The element will only appear as read-only in an ASAP file. **Tags:**atp.EnumerationLiteralIndex=1
readWrite	The element will appear in the ASAP file with both read and write access. **Tags:**atp.EnumerationLiteralIndex=2

Table A.830: SwCalibrationAccessEnum

Class	SwCalprmAxis			
Package	M2::MSR::DataDictionary::CalibrationParameter			
Note	This element specifies an individual input parameter axis (abscissa).			
Base	ARObject			
Aggregated by	SwCalprmAxisSet.swCalprmAxis			
Attribute	Type	Mult.	Kind	Note
category	CalprmAxisCategoryEnum	0..1	attr	This property specifies the category of a particular axis. **Tags:**xml.sequenceOffset=30
displayFormat	DisplayFormatString	0..1	attr	This property specifies how the axis values shall be displayed e.g. in documents or in measurement and calibration tools. **Tags:**xml.sequenceOffset=100
swAxisIndex	AxisIndexType	0..1	attr	This attribute specifies which axis is specified by the containing SwCalprmAxis. For example in a curve this is usually "1". In a map this is "1" or "2". **Tags:**xml.sequenceOffset=20
swCalibrationAccess	SwCalibrationAccessEnum	0..1	attr	Describes the applicability of parameters and variables. **Tags:**xml.sequenceOffset=90
swCalprmAxisTypeProps	SwCalprmAxisTypeProps	0..1	aggr	specific properties depending on the type of the axis. **Tags:** xml.roleElement=false xml.roleWrapperElement=false xml.sequenceOffset=40 xml.typeElement=true xml.typeWrapperElement=false

Table A.831: SwCalprmAxis

Class	SwCalprmAxisSet
Package	M2::MSR::DataDictionary::CalibrationParameter
Note	This element specifies the input parameter axes (abscissas) of parameters (and variables, if these are used adaptively).
Base	ARObject

▽

△

Class	SwCalprmAxisSet			
Aggregated by	SwDataDefProps.swCalprmAxisSet			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
swCalprmAxis	SwCalprmAxis	*	aggr	One axis belonging to this SwCalprmAxisSet
				Tags: xml.roleElement=true xml.roleWrapperElement=false xml.sequenceOffset=20 xml.typeElement=false xml.typeWrapperElement=false

Table A.832: SwCalprmAxisSet

Class	SwCalprmRefProxy			
Package	M2::MSR::DataDictionary::DatadictionaryProxies			
Note	Wrapper class for different kinds of references to a calibration parameter.			
Base	ARObject			
Aggregated by	SwAxisGrouped.swCalprmRef, SwDataDependencyArgs.swCalprmRef			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
arParameter	AutosarParameterRef	0..1	aggr	This represents a Parameter within AUTOSAR. Note that the Datatype of the referenced ParameterDataPrototype shall be an ApplicationDataType of category VALUE.
mcDataInstance	McDataInstance	0..1	ref	This reference is used in the McSupport file to express the final instance of group axis etc. It is not allowed to use this outside of an McDataInstance. The referenced mcDataInstance shall be originated from a ParameterDataPrototype.

Table A.833: SwCalprmRefProxy

Class	SwComponentPrototype			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Composition			
Note	Role of a software component within a composition.			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, CompositionSwComponentType.component			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
type	SwComponentType	0..1	tref	Type of the instance. **Stereotypes:** isOfType

Table A.834: SwComponentPrototype

Class	SwComponentPrototypeAssignment			
Package	M2::AUTOSARTemplates::SystemTemplate::SoftwareCluster			
Note	This meta-class is only required to allow for the variant modeling of an instanceRef. **Tags:**atp.Status=draft			
Base	ARObject			
Aggregated by	CpSoftwareCluster.swComponentAssignment			
Attribute	*Type*	*Mult.*	*Kind*	*Note*

▽

△

Class	SwComponentPrototypeAssignment			
swComponent	SwComponent Prototype	0..1	iref	hierarchical tree(s) of Software Components belonging to this CP Software Cluster. This reference is used to describe the belonging SWCs if the CP Software Cluster is described in the context of a System, **Tags:**atp.Status=draft **InstanceRef implemented by:**ComponentInSystem InstanceRef

Table A.835: SwComponentPrototypeAssignment

Class	*SwComponentType* (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components			
Note	Base class for AUTOSAR software components.			
Base	*ARElement*, ARObject, *AtpBlueprint*, AtpBlueprintable, *AtpClassifier*, *AtpType*, CollectableElement, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	*AtomicSwComponentType*, CompositionSwComponentType, ParameterSwComponentType			
Aggregated by	ARPackage.element			
Attribute	**Type**	*Mult.*	*Kind*	**Note**
consistency Needs	ConsistencyNeeds	*	aggr	This represents the collection of ConsistencyNeeds owned by the enclosing SwComponentType. **Stereotypes:** atpSplitable; 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:** atpSplitable; 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:** atpSplitable; 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:** atpSplitable; 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 A.836: SwComponentType

Class	**SwConnector** (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Composition			
Note	The base class for connectors between ports. Connectors have to be identifiable to allow references from the system constraint template.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	AssemblySwConnector, DelegationSwConnector, PassThroughSwConnector			
Aggregated by	*AtpClassifier*.atpFeature, CompositionSwComponentType.connector			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
mapping	PortInterfaceMapping	0..1	ref	Reference to a PortInterfaceMapping specifying the mapping of unequal named PortInterface elements of the two different PortInterfaces typing the two PortPrototypes which are referenced by the ConnectorPrototype.

Table A.837: SwConnector

Class	<<atpVariation>> **SwDataDefProps**			
Package	M2::MSR::DataDictionary::DataDefProperties			
Note	This class is a collection of properties relevant for data objects under various aspects. One could consider this class as a "pattern of inheritance by aggregation". The properties can be applied to all objects of all classes in which SwDataDefProps is aggregated.			
	Note that not all of the attributes or associated elements are useful all of the time. Hence, the process definition (e.g. expressed with an OCL or a Document Control Instance MSR-DCI) has the task of implementing limitations.			
	SwDataDefProps covers various aspects:			
	• Structure of the data element for calibration use cases: is it a single value, a curve, or a map, but also the recordLayouts which specify how such elements are mapped/converted to the Data Types in the programming language (or in AUTOSAR). This is mainly expressed by properties like swRecordLayout and swCalprmAxisSet			
	• Implementation aspects, mainly expressed by swImplPolicy, swVariableAccessImplPolicy, sw AddrMethod, swPointerTagetProps, baseType, implementationDataType and additionalNative TypeQualifier			
	• Access policy for the MCD system, mainly expressed by swCalibrationAccess			
	• Semantics of the data element, mainly expressed by compuMethod and/or unit, dataConstr, invalidValue			
	• Code generation policy provided by swRecordLayout			
	Tags:vh.latestBindingTime=codeGenerationTime			
Base	*ARObject*			
Aggregated by	*AutosarDataType*.swDataDefProps, CompositeNetworkRepresentation.networkRepresentation, *Data Prototype*.swDataDefProps, DataPrototypeTransformationProps.networkRepresentationProps, DiagnosticDataElement.swDataDefProps, DiagnosticEnvDataElementCondition.swDataDefProps, Dlt Argument.networkRepresentation, FlatInstanceDescriptor.swDataDefProps, ImplementationDataType Element.swDataDefProps, InstantiationDataDefProps.swDataDefProps, ISignal.networkRepresentation Props, McDataInstance.resultingProperties, ParameterAccess.swDataDefProps, PerInstanceMemory.sw DataDefProps, *ReceiverComSpec*.networkRepresentation, *SenderComSpec*.networkRepresentation, SomeipDataPrototypeTransformationProps.networkRepresentation, SwPointerTargetProps.swDataDef Props, SwServiceArg.swDataDefProps, SwSystemconst.swDataDefProps, SystemSignal.physicalProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	<<atpVariation>> **SwDataDefProps**			
additionalNative TypeQualifier	NativeDeclarationString	0..1	attr	This attribute is used to declare native qualifiers of the programming language which can neither be deduced from the baseType (e.g. because the data object describes a pointer) nor from other more abstract attributes. Examples are qualifiers like "volatile", "strict" or "enum" of the C-language. All such declarations have to be put into one string. **Tags:**xml.sequenceOffset=235
annotation	Annotation	*	aggr	This aggregation allows to add annotations (yellow pads ...) related to the current data object. **Tags:** xml.roleElement=true xml.roleWrapperElement=true xml.sequenceOffset=20 xml.typeElement=false xml.typeWrapperElement=false
baseType	SwBaseType	0..1	ref	Base type associated with the containing data object. **Tags:**xml.sequenceOffset=50
compuMethod	CompuMethod	0..1	ref	Computation method associated with the semantics of this data object. **Tags:**xml.sequenceOffset=180
dataConstr	DataConstr	0..1	ref	Data constraint for this data object. **Tags:**xml.sequenceOffset=190
displayFormat	DisplayFormatString	0..1	attr	This property describes how a number is to be rendered e.g. in documents or in a measurement and calibration system. **Tags:**xml.sequenceOffset=210
display Presentation	DisplayPresentation Enum	0..1	attr	This attribute controls the presentation of the related data for measurement and calibration tools.
implementation DataType	AbstractImplementation DataType	0..1	ref	This association denotes the ImplementationDataType of a data declaration via its aggregated SwDataDefProps. It is used whenever a data declaration is not directly referring to a base type. Especially - redefinition of an ImplementationDataType via a "typedef" to another ImplementationDatatype - the target type of a pointer (see SwPointerTarget Props), if it does not refer to a base type directly - the data type of an array or record element within an ImplementationDataType, if it does not refer to a base type directly - the data type of an SwServiceArg, if it does not refer to a base type directly **Tags:**xml.sequenceOffset=215
invalidValue	ValueSpecification	0..1	aggr	Optional value to express invalidity of the actual data element. **Tags:**xml.sequenceOffset=255
stepSize	Float	0..1	attr	This attribute can be used to define a value which is added to or subtracted from the value of a DataPrototype when using up/down keys while calibrating.

▽

△

Class	<<atpVariation>> **SwDataDefProps**			
swAddrMethod	SwAddrMethod	0..1	ref	Addressing method related to this data object. Via an association to the same SwAddrMethod it can be specified that several DataPrototypes shall be located in the same memory without already specifying the memory section itself. **Tags:**xml.sequenceOffset=30
swAlignment	AlignmentType	0..1	attr	The attribute describes the intended typical alignment of the DataPrototype. If the attribute is not defined the alignment is determined by the swBaseType size and the memoryAllocationKeywordPolicy of the referenced Sw AddrMethod. **Tags:**xml.sequenceOffset=33
swBit Representation	SwBitRepresentation	0..1	aggr	Description of the binary representation in case of a bit variable. **Tags:**xml.sequenceOffset=60
swCalibration Access	SwCalibrationAccess Enum	0..1	attr	Specifies the read or write access by MCD tools for this data object. **Tags:**xml.sequenceOffset=70
swCalprmAxis Set	SwCalprmAxisSet	0..1	aggr	This specifies the properties of the axes in case of a curve or map etc. This is mainly applicable to calibration parameters. **Tags:**xml.sequenceOffset=90
swComparison Variable	SwVariableRefProxy	*	aggr	Variables used for comparison in an MCD process. **Tags:** xml.sequenceOffset=170 xml.typeElement=false
swData Dependency	SwDataDependency	0..1	aggr	Describes how the value of the data object has to be calculated from the value of another data object (by the MCD system). **Tags:**xml.sequenceOffset=200
swHostVariable	SwVariableRefProxy	0..1	aggr	Contains a reference to a variable which serves as a host-variable for a bit variable. Only applicable to bit objects. **Tags:** xml.sequenceOffset=220 xml.typeElement=false
swImplPolicy	SwImplPolicyEnum	0..1	attr	Implementation policy for this data object. **Tags:**xml.sequenceOffset=230
swIntended Resolution	Numerical	0..1	attr	The purpose of this element is to describe the requested quantization of data objects early on in the design process. The resolution ultimately occurs via the conversion formula present (compuMethod), which specifies the transition from the physical world to the standardized world (and vice-versa) (here, "the slope per bit" is present implicitly in the conversion formula). In the case of a development phase without a fixed conversion formula, a pre-specification can occur through swIntendedResolution. The resolution is specified in the physical domain according to the property "unit". **Tags:**xml.sequenceOffset=240

▽

Document ID 635: AUTOSAR_TR_ClassicAutosarModelConstraints

△

Class	<<atpVariation>> **SwDataDefProps**			
swInterpolation Method	Identifier	0..1	attr	This is a keyword identifying the mathematical method to be applied for interpolation. The keyword needs to be related to the interpolation routine which needs to be invoked. **Tags:**xml.sequenceOffset=250
swIsVirtual	Boolean	0..1	attr	This element distinguishes virtual objects. Virtual objects do not appear in the memory, their derivation is much more dependent on other objects and hence they shall have a swDataDependency . **Tags:**xml.sequenceOffset=260
swPointerTarget Props	SwPointerTargetProps	0..1	aggr	Specifies that the containing data object is a pointer to another data object. **Tags:**xml.sequenceOffset=280
swRecord Layout	SwRecordLayout	0..1	ref	Record layout for this data object. **Tags:**xml.sequenceOffset=290
swRefresh Timing	MultidimensionalTime	0..1	aggr	This element specifies the frequency in which the object involved shall be or is called or calculated. This timing can be collected from the task in which write access processes to the variable run. But this cannot be done by the MCD system. So this attribute can be used in an early phase to express the desired refresh timing and later on to specify the real refresh timing. **Tags:**xml.sequenceOffset=300
swTextProps	SwTextProps	0..1	aggr	the specific properties if the data object is a text object. **Tags:**xml.sequenceOffset=120
swValueBlock Size	Numerical	0..1	attr	This represents the size of a Value Block **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=80
swValueBlock SizeMult (ordered)	Numerical	*	attr	This attribute is used to specify the dimensions of a value block (VAL_BLK) for the case that that value block has more than one dimension. The dimensions given in this attribute are ordered such that the first entry represents the first dimension, the second entry represents the second dimension, and so on. For one-dimensional value blocks the attribute swValue BlockSize shall be used and this attribute shall not exist. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime
unit	Unit	0..1	ref	Physical unit associated with the semantics of this data object. This attribute applies if no compuMethod is specified. If both units (this as well as via compuMethod) are specified the units shall be compatible. **Tags:**xml.sequenceOffset=350
valueAxisData Type	ApplicationPrimitive DataType	0..1	ref	The referenced ApplicationPrimitiveDataType represents the primitive data type of the value axis within a compound primitive (e.g. curve, map). It supersedes CompuMethod, Unit, and BaseType. **Tags:**xml.sequenceOffset=355

Table A.838: SwDataDefProps

Class	SwGenericAxisParam
Package	M2::MSR::DataDictionary::Axis
Note	This meta-class describes a specific parameter of a generic axis. The name of the parameter is defined through a reference to a parameter type defined on a corresponding axis type. The value of the parameter is given here in case that it is not changeable during calibration. Example is shift / offset in a fixed axis.
Base	ARObject
Aggregated by	SwAxisGeneric.swGenericAxisParam

Attribute	Type	Mult.	Kind	Note
swGenericAxis ParamType	SwGenericAxisParam Type	0..1	ref	Parameter type defined on a corresponding axis type. References can only be made to axis parameters types which are defined within the referenced axis type. **Tags:**xml.sequenceOffset=20
vf (ordered)	Numerical	*	attr	This attribute represents the value of the generic axis parameter. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.roleElement=true xml.roleWrapperElement=false xml.sequenceOffset=30 xml.typeElement=false

Table A.839: SwGenericAxisParam

Class	SwGenericAxisParamType
Package	M2::MSR::DataDictionary::Axis
Note	This meta-class describes a generic axis parameter type, namely: • Plausibility checks can be specified via dataConstr. • Textual description (desc), as a formal description is not of any use, due to the large variety of possibilities. • If this parameter contains structures, these can be simulated through the recursive use of Sw GenericAxisParamTypes.
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable
Aggregated by	SwAxisType.swGenericAxisParamType

Attribute	Type	Mult.	Kind	Note
dataConstr	DataConstr	0..1	ref	This reference denoted data constraints applicable to the generic axis parameter. **Tags:**xml.sequenceOffset=20

Table A.840: SwGenericAxisParamType

Enumeration	SwImplPolicyEnum
Package	M2::MSR::DataDictionary::DataDefProperties
Note	Specifies the implementation strategy with respect to consistency mechanisms of variables.
Aggregated by	BswInternalTriggeringPoint.swImplPolicy, InternalTriggeringPoint.swImplPolicy, SwDataDefProps.sw ImplPolicy, Trigger.swImplPolicy
Literal	**Description**

$\bigtriangledown$

△

Enumeration	SwImplPolicyEnum
const	forced implementation such that the running software within the ECU shall not modify it. For example implemented with the "const" modifier in C. This can be applied for parameters (not for those in NVRAM) as well as argument data prototypes. **Tags:**atp.EnumerationLiteralIndex=0
fixed	This data element is fixed. In particular this indicates, that it might also be implemented e.g. as in place data, (#DEFINE). **Tags:**atp.EnumerationLiteralIndex=1
measurementPoint	The data element is created for measurement purposes only. The data element is never read directly within the ECU software. In contrast to a "standard" data element in an unconnected provide port is, this unconnection is guaranteed for measurementPoint data elements. **Tags:**atp.EnumerationLiteralIndex=2
queued	The content of the data element is queued and the data element has 'event' semantics, i.e. data elements are stored in a queue and all data elements are processed in 'first in first out' order. The queuing is intended to be implemented by RTE Generator. This value is not applicable for parameters. **Tags:**atp.EnumerationLiteralIndex=3
standard	This is applicable for all kinds of data elements. For variable data prototypes the 'last is best' semantics applies. For parameter there is no specific implementation directive. **Tags:**atp.EnumerationLiteralIndex=4

Table A.841: SwImplPolicyEnum

Class	SwPointerTargetProps			
Package	M2::MSR::DataDictionary::DataDefProperties			
Note	This element defines, that the data object (which is specified by the aggregating element) contains a reference to another data object or to a function in the CPU code. This corresponds to a pointer in the C-language. The attributes of this element describe the category and the detailed properties of the target which is either a data description or a function signature.			
Base	ARObject			
Aggregated by	SwDataDefProps.swPointerTargetProps			
Attribute	Type	Mult.	Kind	Note
functionPointer Signature	BswModuleEntry	0..1	ref	The referenced BswModuleEntry serves as the signature of a function pointer definition. Primary use case: function pointer passed as argument to other function. **Tags:**xml.sequenceOffset=40
swDataDef Props	SwDataDefProps	0..1	aggr	The properties of the target data type. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=swDataDefProps xml.sequenceOffset=30
targetCategory	Identifier	0..1	attr	This specifies the category of the target: • In case of a data pointer, it shall specify the category of the referenced data. • In case of a function pointer, it could be used to denote the category of the referenced Bsw ModuleEntry. **Tags:**xml.sequenceOffset=5

Table A.842: SwPointerTargetProps

Class	SwSystemconst			
Package	M2::MSR::DataDictionary::SystemConstant			
Note	This element defines a system constant which serves an input to select a particular variation point. In particular a system constant serves as an operand of the binding function (swSyscond) in a Variation point. Note that the binding process can only happen if a value was assigned to to the referenced system constants. **Tags:**atp.recommendedPackage=SwSystemconsts			
Base	*ARElement*, *ARObject*, *AtpDefinition*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
swDataDef Props	SwDataDefProps	0..1	aggr	This denotes the data definition properties of the system constant. This supports to express the limits and optionally a conversion within the internal to physical values by a compu method. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=swDataDefProps xml.sequenceOffset=40

Table A.845: SwSystemconst

Class	SwSystemconstValue			
Package	M2::AUTOSARTemplates::GenericStructure::VariantHandling			
Note	This meta-class assigns a particular value to a system constant.			
Base	*ARObject*			
Aggregated by	SwSystemconstantValueSet.swSystemconstantValue			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
annotation	Annotation	*	aggr	This provides the ability to add information why the value is set like it is. **Tags:**xml.sequenceOffset=30
swSystemconst	SwSystemconst	1	ref	This is the system constant to which the value applies. **Tags:**xml.sequenceOffset=10
value	Numerical	1	attr	This is the particular value of a system constant. It is specified as Numerical. Further restrictions may apply by the definition of the system constant. The value attribute defines the internal value of the Sw Systemconst as it is processed in the Formula Language. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=20

Table A.846: SwSystemconstValue

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*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*

▽

△

Class	SwSystemconstantValueSet			
Aggregated by	ARPackage.element			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
sw Systemconstant Value	SwSystemconstValue	*	aggr	This is one particular value of a system constant.

Table A.847: SwSystemconstantValueSet

Class	SwTextProps			
Package	M2::MSR::DataDictionary::DataDefProperties			
Note	This meta-class expresses particular properties applicable to strings in variables or calibration parameters.			
Base	ARObject			
Aggregated by	SwDataDefProps.swTextProps			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
arraySize Semantics	ArraySizeSemantics Enum	0..1	attr	This attribute controls the semantics of the arraysize for the array representing the string in an Implementation DataType. It is there to support a safe conversion between ApplicationDatatype and ImplementationDatatype, even for variable length strings as required e.g. for Support of SAE J1939.
baseType	SwBaseType	0..1	ref	This is the base type of one character in the string. In particular this baseType denotes the intended encoding of the characters in the string on level of ApplicationData Type. **Tags:**xml.sequenceOffset=30
swFillCharacter	Integer	0..1	attr	Filler character for text parameter to pad up to the maximum length swMaxTextSize. The value will be interpreted according to the encoding specified in the associated base type of the data object, e.g. 0x30 (hex) represents the ASCII character zero as filler character and 0 (dec) represents an end of string as filler character. The usage of the fill character depends on the arraySize Semantics. **Tags:**xml.sequenceOffset=40
swMaxTextSize	Integer	0..1	attr	Specifies the maximum text size in characters. Note the size in bytes depends on the encoding in the corresponding baseType. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=20

Table A.848: SwTextProps

Class	SwValueCont
Package	M2::MSR::CalibrationData::CalibrationValue
Note	This metaclass represents the content of one particular SwInstance.
Base	ARObject

▽

△

Class	SwValueCont			
Aggregated by	ApplicationValueSpecification.swValueCont			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
swArraysize	ValueList	0..1	aggr	This attribute defines the size of each dimension for compound primitives CURVE, MAP, CUBOID, CUBE_4, CUBE_5, COM_AXIS, RES_AXIS, VAL_BLK. For each dimension one value has to be defined, e.g. one in case of COM_AXIS and two or more in case of MAP. **Tags:**xml.sequenceOffset=40
swValuesPhys	SwValues	0..1	aggr	swValuesPhys represents the values in the physical domain. **Tags:**xml.sequenceOffset=50
unit	Unit	0..1	ref	This represents the physical unit of the provided values. **Tags:**xml.sequenceOffset=20
unitDisplay Name	SingleLanguageUnit Names	0..1	aggr	This specifies how the physical units of the current value set shall be displayed in documents or in user interfaces of tools. **Tags:**xml.sequenceOffset=30

Table A.849: SwValueCont

Class	<<atpMixed>> **SwValues**			
Package	M2::MSR::CalibrationData::CalibrationValue			
Note	This meta-class represents a list of values. These values can either be the input values of a curve (abscissa values) or the associated values (ordinate values). In case of multidimensional structures, the values are ordered such that the lowest index runs the fastest. In particular for maps and cuboids etc. the resulting long value list can be subsectioned using Value Group. But the processing needs to be done as if vg is not there. Note that numerical values and textual values should not be mixed.			
Base	ARObject			
Aggregated by	SwAxisCont.swValuesPhys, SwValueCont.swValuesPhys, ValueGroup.vgContents			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
v	Numerical	0..1	attr	This is a non variant Value. It is provided for sake of Compatibility to ASAM CDF. **Tags:**xml.sequenceOffset=40
vf	Numerical	0..1	attr	This allows to specify the value as VariationPoint. It is distinguished to non variant for sake of compatibility to ASAM CDF 2.0. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=20
vg	ValueGroup	0..1	aggr	This allows to have intersections in the values in order to support specific rendering (eg. using stylesheets). For tools it is important that the v values are always processed in the same (flattened) order and the tool is able to interpret it without respecting vg. **Tags:**xml.sequenceOffset=50

▽

△

Class	<<atpMixed>> **SwValues**			
vt	VerbatimString	0..1	attr	This represents the values of textual data elements (Strings). Note that vt uses the | to separate the values for the different bitfield masks in case that the semantics of the related DataPrototype is described by means of a BITFIELD_TEXTTABLE in the associated CompuMethod. **Tags:**xml.sequenceOffset=30
vtf	NumericalOrText	0..1	aggr	This aggregation represents the ability to provide a value that is either numerical or text which existence is subject to variability. From the formal point of view, the aggregation needs to have the multiplicity 1 because SwValues is modelled with stereotype <<atpMixed>>. Nevertheless, the existence of vtf is optional and subject to constraints. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=vtf, vtf.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table A.850: SwValues

Class	**SwVariableRefProxy**			
Package	M2::MSR::DataDictionary::DatadictionaryProxies			
Note	Proxy class for several kinds of references to a variable.			
Base	ARObject			
Aggregated by	SwAxisIndividual.swVariableRef, SwDataDefProps.swComparisonVariable, SwDataDefProps.swHostVariable, SwDataDependencyArgs.swVariable			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
autosarVariable	AutosarVariableRef	0..1	aggr	This represents the reference to a Variable in an Autosar system. Note that the target of the reference within AutosarVariableRef shall be typed by a primitive data type
mcDataInstanceVar	McDataInstance	0..1	ref	This reference is used in the McSupport file to express the final instance of input values etc. It is not allowed to use this outside of an McDataInstance. The referenced mcDataInstance shall be originated from a VariableDataPrototype.

Table A.851: SwVariableRefProxy

Class	**SwcBswMapping**			
Package	M2::AUTOSARTemplates::CommonStructure::SwcBswMapping			
Note	Maps an SwcInternalBehavior to an BswInternalBehavior. This is required to coordinate the API generation and the scheduling for AUTOSAR Service Components, ECU Abstraction Components and Complex Driver Components by the RTE and the BSW scheduling mechanisms. **Tags:**atp.recommendedPackage=SwcBswMappings			
Base	*ARElement*, *ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element, *AtpClassifier*.atpFeature			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
bswBehavior	BswInternalBehavior	0..1	ref	The mapped BswInternalBehavior

▽

△

Class	SwcBswMapping			
runnable Mapping	SwcBswRunnable Mapping	*	aggr	A mapping between a pair of SWC and BSW runnables. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=runnableMapping, runnable Mapping.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
swcBehavior	SwcInternalBehavior	0..1	ref	The mapped SwcInternalBehavior.
synchronized ModeGroup	SwcBswSynchronized ModeGroupPrototype	*	aggr	A pair of SWC and BSW mode group prototypes to be synchronized by the scheduler. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=synchronizedModeGroup, synchronized ModeGroup.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
synchronized Trigger	SwcBswSynchronized Trigger	*	aggr	A pair of SWC and BSW Triggers to be synchronized by the scheduler. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=synchronizedTrigger, synchronized Trigger.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table A.852: SwcBswMapping

Class	SwcBswRunnableMapping			
Package	M2::AUTOSARTemplates::CommonStructure::SwcBswMapping			
Note	Maps a BswModuleEntity to a RunnableEntity if it is implemented as part of a BSW module (in the case of an AUTOSAR Service, a Complex Driver or an ECU Abstraction). The mapping can be used by a tool to find relevant information on the behavior, e.g. whether the bswEntity shall be running in interrupt context.			
Base	ARObject			
Aggregated by	SwcBswMapping.runnableMapping			
Attribute	Type	Mult.	Kind	Note
bswEntity	BswModuleEntity	0..1	ref	The mapped BswModuleEntity
swcRunnable	RunnableEntity	0..1	ref	The mapped SWC runnable.

Table A.853: SwcBswRunnableMapping

Class	SwcBswSynchronizedModeGroupPrototype			
Package	M2::AUTOSARTemplates::CommonStructure::SwcBswMapping			
Note	Synchronizes a mode group provided by a component via a port with a mode group provided by a BSW module or cluster.			
Base	ARObject			
Aggregated by	SwcBswMapping.synchronizedModeGroup			
Attribute	Type	Mult.	Kind	Note
bswModeGroup	ModeDeclarationGroup Prototype	0..1	ref	The BSW mode group prototype.
swcModeGroup	ModeDeclarationGroup Prototype	0..1	iref	The SWC mode group prototype provided by a particular port. **InstanceRef implemented by:**PModeGroupInAtomic SwcInstanceRef

Table A.854: SwcBswSynchronizedModeGroupPrototype

Class	SwcBswSynchronizedTrigger			
Package	M2::AUTOSARTemplates::CommonStructure::SwcBswMapping			
Note	Synchronizes a Trigger provided by a component via a port with a Trigger provided by a BSW module or cluster.			
Base	ARObject			
Aggregated by	SwcBswMapping.synchronizedTrigger			
Attribute	Type	Mult.	Kind	Note
bswTrigger	Trigger	0..1	ref	The BSW Trigger.
swcTrigger	Trigger	0..1	iref	The SWC Trigger provided by a particular port. **InstanceRef implemented by:**PTriggerInAtomicSwcTypeInstanceRef

Table A.855: SwcBswSynchronizedTrigger

Class	SwcExclusiveAreaPolicy			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior			
Note	Options how to generate the ExclusiveArea related APIs. If no SwcExclusiveAreaPolicy is specified for an ExclusiveArea the default values apply.			
Base	ARObject			
Aggregated by	SwcInternalBehavior.exclusiveAreaPolicy			
Attribute	Type	Mult.	Kind	Note
apiPrinciple	ApiPrincipleEnum	0..1	attr	Specifies for this ExclusiveArea if either one common set of Enter and Exit APIs for the whole software component is requested from the Rte or if the set of Enter and Exit APIs is expected per RunnableEntity. The default value is "common".
exclusiveArea	ExclusiveArea	0..1	ref	This reference represents the ExclusiveArea for which the policy applies.

Table A.856: SwcExclusiveAreaPolicy

Class	SwcImplementation			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcImplementation			
Note	This meta-class represents a specialization of the general Implementation meta-class with respect to the usage in application software. **Tags:**atp.recommendedPackage=SwcImplementations			
Base	ARElement, ARObject, CollectableElement, Identifiable, Implementation, MultilanguageReferrable, PackageableElement, Referrable			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
behavior	SwcInternalBehavior	0..1	ref	The internal behavior implemented by this Implementation.

$\bigtriangledown$

△

Class	SwcImplementation			
perInstance MemorySize	PerInstanceMemory Size	*	aggr	Allows a definition of the size of the per-instance memory for this implementation. The aggregation of PerInstance MemorySize 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, in this case PerInstanceMemory. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=perInstanceMemorySize, perInstance MemorySize.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
required RTEVendor	String	0..1	attr	Identify a specific RTE vendor. This information is potentially important at the time of integrating (in particular: linking) the application code with the RTE. The semantics is that (if the association exists) the corresponding code has been created to fit to the vendor-mode RTE provided by this specific vendor. Attempting to integrate the code with another RTE generated in vendor mode is in general not possible.

Table A.857: SwcImplementation

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, Identifiable, InternalBehavior, Multilanguage Referrable, Referrable			
Aggregated by	AtomicSwComponentType.internalBehavior, AtpClassifier.atpFeature			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
arTypedPer Instance Memory	VariableDataPrototype	*	aggr	Defines an AUTOSAR typed memory-block that needs to be available for each instance of the SW-component. This is typically only useful if supportsMultipleInstantiation is set to "true" or if the component defines NVRAM access via permanent blocks. 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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=arTypedPerInstanceMemory.shortName, ar TypedPerInstanceMemory.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	**SwcInternalBehavior**			
event	RTEEvent	*	aggr	This is a RTEEvent specified for the particular Swc InternalBehavior. 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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=event.shortName, event.variationPoint.short Label vh.latestBindingTime=preCompileTime
exclusiveArea Policy	SwcExclusiveArea Policy	*	aggr	Options how to generate the ExclusiveArea related APIs. When no SwcExclusiveAreaPolicy is specified for an ExclusiveArea the default values apply. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=exclusiveAreaPolicy, exclusiveArea Policy.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
explicitInter Runnable Variable	VariableDataPrototype	*	aggr	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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=explicitInterRunnableVariable.shortName, explicitInterRunnableVariable.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
handle TerminationAnd Restart	HandleTerminationAnd RestartEnum	0..1	attr	This attribute controls the behavior with respect to stopping and restarting. The corresponding AtomicSw ComponentType may either not support stop and restart, or support only stop, or support both stop and restart.
implicitInter Runnable Variable	VariableDataPrototype	*	aggr	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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=implicitInterRunnableVariable.shortName, implicitInterRunnableVariable.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
includedData TypeSet	IncludedDataTypeSet	*	aggr	The includedDataTypeSet is used by a software component for its implementation. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=includedDataTypeSet

▽

△

Class	**SwcInternalBehavior**			
includedMode Declaration GroupSet	IncludedMode DeclarationGroupSet	*	aggr	This aggregation represents the included Mode DeclarationGroups **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=includedModeDeclarationGroupSet
instantiation DataDefProps	InstantiationDataDef Props	*	aggr	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". **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=instantiationDataDefProps, instantiationData DefProps.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
perInstance Memory	PerInstanceMemory	*	aggr	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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=perInstanceMemory.shortName, perInstance Memory.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
perInstance Parameter	ParameterData Prototype	*	aggr	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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=perInstanceParameter.shortName, per InstanceParameter.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
portAPIOption	PortAPIOption	*	aggr	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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=portAPIOption, portAPIOption.variation Point.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	SwcInternalBehavior			
runnable	RunnableEntity	*	aggr	This is a RunnableEntity specified for the particular Swc InternalBehavior. 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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=runnable.shortName, runnable.variation Point.shortLabel vh.latestBindingTime=preCompileTime
service Dependency	SwcService Dependency	*	aggr	Defines the requirements on AUTOSAR Services for a particular item. 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. The SwcServiceDependency owned by an SwcInternal 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>>. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=serviceDependency.shortName, service Dependency.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
shared Parameter	ParameterData Prototype	*	aggr	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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=sharedParameter.shortName, shared Parameter.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
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	Proxy of a variation points in the C/C++ implementation. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=variationPointProxy.shortName

Table A.858: SwcInternalBehavior

Class	SwcModeManagerErrorEvent			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents			
Note	This event is raised when an error occurred during the handling of the referenced ModeDeclarationGroup Prototype.			
Base	*ARObject*, *AbstractEvent*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *Multilanguage Referrable*, *RTEEvent*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, SwcInternalBehavior.event			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
modeGroup	ModeDeclarationGroup Prototype	0..1	iref	This represents the ModeDeclarationGroupPrototype for which this SwcModeManagerErrorEvent is raised in case of an error. **InstanceRef implemented by:**PModeGroupInAtomic SwcInstanceRef

Table A.859: SwcModeManagerErrorEvent

Class	SwcModeSwitchEvent			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents			
Note	This event is raised when the specified mode change occurs.			
Base	*ARObject*, *AbstractEvent*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *Multilanguage Referrable*, *RTEEvent*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, SwcInternalBehavior.event			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
activation	ModeActivationKind	0..1	attr	Specifies if the event is raised on entering or exiting a specific mode or is raised on the transition between two modes.
mode (ordered)	ModeDeclaration	0..2	iref	The referenced mode or the transition between two modes raises this SwcModeSwitchEvent. **InstanceRef implemented by:**RModeInAtomicSwc InstanceRef

Table A.860: SwcModeSwitchEvent

Class	SwcServiceDependency			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::ServiceMapping			
Note	Specialization of ServiceDependency in the context of an SwcInternalBehavior. It allows to associate ports, port groups and (in special cases) data defined for an atomic software component to a given ServiceNeeds element.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *ServiceDependency*			
Aggregated by	AdaptiveSwcInternalBehavior.serviceDependency, *AtpClassifier*.atpFeature, SwcInternalBehavior. serviceDependency			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
assignedData	RoleBasedData Assignment	*	aggr	Defines the role of an associated data object of the same component. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=assignedData, assignedData.variation Point.shortLabel vh.latestBindingTime=preCompileTime

$\triangledown$

△

Class	SwcServiceDependency			
assignedPort	RoleBasedPort Assignment	*	aggr	Defines the role of an associated port of the same component. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=assignedPort, assignedPort.variation Point.shortLabel vh.latestBindingTime=preCompileTime
representedPort Group	PortGroup	0..1	ref	This reference specifies an association between the ServiceNeeeds and a PortGroup, for example to request a communication mode which applies for communication via these ports. The referred PortGroup shall be local to this atomic SWC, but via the links between the Port Groups, a tool can evaluate this information such that all the ports linked via this port group on the same ECU can be found.
serviceNeeds	ServiceNeeds	0..1	aggr	The associated ServiceNeeds.

Table A.861: SwcServiceDependency

Class	SwcToApplicationPartitionMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::SWmapping			
Note	Allows to map a given SwComponentPrototype to a formally defined partition at a point in time when the corresponding EcuInstance is not yet known or defined.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	CpSoftwareClusterMappingSet.swcToApplicationPartitionMapping, SwComponentMappingConstraints. swcToApplicationPartitionMapping, SystemMapping.swcToApplicationPartitionMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
application Partition	ApplicationPartition	0..1	ref	Reference to an ApplicationPartition to which a Sw ComponentPrototype is mapped.
swComponent Prototype	SwComponent Prototype	0..1	iref	References to the software component instances that are mapped to the referenced ApplicationPartition. If the component prototype referenced is a composition, this indicates that all atomic software components within the composition are mapped to the ApplicationPartition. If there is additionally a mapping of some SwComponent Prototype INSIDE the Composition to another Application Partition the inner mapping overrides the outer mapping. **InstanceRef implemented by:**ComponentInSystem InstanceRef

Table A.862: SwcToApplicationPartitionMapping

Class	SwcToEcuMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::SWmapping			
Note	Map software components to a specific ECU Instance and optionally to a processing unit and to an Ecu Partition. For each combination of ECUInstance and the optional ProcessingUnit and the optional Ecu Partition and the optional SensorActuator only one SwcToEcuMapping shall be used.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	SystemMapping.swMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	SwcToEcuMapping			
component	SwComponent Prototype	1..*	iref	References to the software component instances that are mapped to the referenced ECUInstance. If the component prototype referenced is a composition, this indicates that all atomic software components within the composition are mapped to the ECU. If there is aditionally a mapping of some SwComponent Prototype INSIDE the Composition to another ECU Instance the inner mapping overrides the outer mapping. **InstanceRef implemented by:**ComponentInSystem InstanceRef
controlledHw Element	HwElement	0..1	ref	Optional mapping of SwComponentPrototypes that are typed by SensorActuatorSwComponentType to a Hw Element with category SensorActuator.
ecuInstance	EcuInstance	1	ref	Reference to a specific ECU Instance description.
processingUnit	HwElement	0..1	ref	Optional mapping of software components to individual microcontroller cores residing in one ECU. A microcontroller core is described in the ECU Resource Template by the HwElement of HwCategory Processing Unit.

Table A.863: SwcToEcuMapping

Class	SymbolProps			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components			
Note	This meta-class represents the ability to attach with the symbol attribute a symbolic name that is conform to C language requirements to another meta-class, e.g. AtomicSwComponentType, that is a potential subject to a name clash on the level of RTE source code.			
Base	*ARObject*, *ImplementationProps*, *Referrable*			
Aggregated by	Allocator.namespace, ApApplicationErrorDomain.namespace, *AtomicSwComponentType*.symbolProps, *CppImplementationDataType*.namespace, ImplementationDataType.symbolProps, *PortInterface*. namespace, SecurityEventDefinition.eventSymbolName			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
–	–	–	–	–

Table A.864: SymbolProps

Class	SynchronizationTimingConstraint			
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingConstraint::SynchronizationTiming Constraint			
Note	This constraint is used to restrict the timing behavior of different, but correlated events or event chains, with regard to synchronization. Thereby, in case of imposing a synchronization timing constraint on events or event chains the following two scenarios are supported: 1) [synchronizationConstraintType=responseSynchronization] Events: An arbitrary number of correlated events which play the role of responses shall occur synchronously with respect to a predefined tolerance. Event Chains: 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. 2) [synchronizationConstraintType=stimulusSynchronization] Events: An arbitrary number of correlated events which play the role of stimuli shall occur synchronously with respect to a predefined tolerance. Event Chains: 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. In case of imposing a synchronization timing constraint on events the following two scenarios are supported: 1) [eventOccurrenceKind=singleOccurrence] Any of the events shall occur only once in the given time interval. 2) [eventOccurrenceKind=multipleOccurrences] 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	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TimingConstraint*, *Traceable*			
Aggregated by	*TimingExtension*.timingGuarantee, *TimingExtension*.timingRequirement			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
event OccurrenceKind	EventOccurrenceKind Enum	0..1	attr	The specific occurrence kind of an event occurring within the given time interval.
scope	TimingDescriptionEvent Chain	*	ref	The event chains that are in the scope of the constraint.
scopeEvent	TimingDescriptionEvent	*	ref	The events that are in the scope of the constraint.
synchronization ConstraintType	SynchronizationType Enum	0..1	attr	The specific type of this synchronization constraint.
tolerance	MultidimensionalTime	0..1	aggr	The maximum time interval, within which the synchronized events shall occur.

Table A.865: SynchronizationTimingConstraint

Class	SynchronousServerCallPoint			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::ServerCall			
Note	This means that the RunnableEntity is supposed to perform a blocking wait for a response from the server.			
Base	*ARObject*, *AbstractAccessPoint*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *ServerCallPoint*			
Aggregated by	*AtpClassifier*.atpFeature, RunnableEntity.serverCallPoint			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
calledFrom WithinExclusive Area	ExclusiveAreaNesting Order	0..1	ref	This indicates that the call point is located at the deepest level inside one or more ExclusiveAreas that are nested in the given order.

Table A.866: SynchronousServerCallPoint

Class	System			
Package	M2::AUTOSARTemplates::SystemTemplate			
Note	The top level element of the System Description. The System description defines five major elements: Topology, Software, Communication, Mapping and Mapping Constraints.			
	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.			
	Tags: atp.recommendedPackage=Systems			
Base	*ARElement*, *ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Aggregated by	ARPackage.element, *AtpClassifier*.atpFeature			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
clientId DefinitionSet	ClientIdDefinitionSet	*	ref	Set of Client Identifiers that are used for inter-ECU client-server communication in the System.
containerIPdu HeaderByte Order	ByteOrderEnum	0..1	attr	Defines the byteOrder of the header in ContainerIPdus.
ecuExtract Version	RevisionLabelString	0..1	attr	Version number of the Ecu Extract.
fibexElement	FibexElement	*	ref	Reference to ASAM FIBEX elements specifying Communication and Topology.
				All Fibex Elements used within a System Description shall be referenced from the System Element.
				atpVariation: In order to describe a product-line, all Fibex Elements can be optional.
				Stereotypes: atpSplitable; atpVariation **Tags:** atp.Splitkey=fibexElement.fibexElement, fibex Element.variationPoint.shortLabel vh.latestBindingTime=postBuild
interpolation Routine MappingSet	InterpolationRoutine MappingSet	*	ref	This reference identifies the InterpolationRoutineMapping Sets that are relevant in the context of the enclosing System.
j1939Shared AddressCluster	J1939SharedAddress Cluster	*	aggr	Collection of J1939Clusters that share a common address space for the routing of messages.
				Stereotypes: atpSplitable; atpVariation **Tags:** atp.Splitkey=j1939SharedAddressCluster.shortName, j1939SharedAddressCluster.variationPoint.shortLabel vh.latestBindingTime=postBuild
mapping	SystemMapping	*	aggr	Aggregation of all mapping aspects (mapping of SW components to ECUs, mapping of data elements to signals, and mapping constraints).
				In order to support OEM / Tier 1 interaction and shared development for one common System this aggregation is atpSplitable and atpVariation. The content of System Mapping can be provided by several parties using different names for the SystemMapping.
				This element is not required when the System description is used for a network-only use-case.
				Stereotypes: atpSplitable; atpVariation **Tags:** atp.Splitkey=mapping.shortName, mapping.variation Point.shortLabel vh.latestBindingTime=postBuild
pncVector Length	PositiveInteger	0..1	attr	Length of the partial networking request release information vector (in bytes).

▽

△

Class	System			
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.
rootSoftware Composition	RootSwComposition Prototype	0..1	aggr	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. atpVariation: The RootSwCompositionPrototype can vary. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=rootSoftwareComposition.shortName, root SoftwareComposition.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime
swCluster	CpSoftwareCluster	*	ref	CP Software Clusters of this System **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=swCluster.cpSoftwareCluster, sw Cluster.variationPoint.shortLabel atp.Status=draft vh.latestBindingTime=systemDesignTime
system Documentation	Chapter	*	aggr	Possibility to provide additional documentation while defining the System. The System documentation can be composed of several chapters. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=systemDocumentation.shortName, system Documentation.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=-10
systemVersion	RevisionLabelString	1	attr	Version number of the System Description.

Table A.867: System

Class	SystemMapping			
Package	M2::AUTOSARTemplates::SystemTemplate			
Note	The system mapping aggregates all mapping aspects (mapping of SW components to ECUs, mapping of data elements to signals, and mapping constraints).			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	System.mapping			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
application PartitionToEcu Partition Mapping	ApplicationPartitionTo EcuPartitionMapping	*	aggr	Mapping of ApplicationPartitions to EcuPartitions **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=applicationPartitionToEcuPartition Mapping.shortName, applicationPartitionToEcuPartition Mapping.variationPoint.shortLabel vh.latestBindingTime=postBuild
appOsTask ProxyToEcu TaskProxy Mapping	AppOsTaskProxyToEcu TaskProxyMapping	*	aggr	Mapping of an OsTaskProxy that was created in the context of a SwComponent to an OsTaskProxy that was created in the context of an Ecu.

▽

△

Class	**SystemMapping**			
com Management Mapping	ComManagement Mapping	*	aggr	Mappings between Mode Management PortGroups and communication channels. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=comManagementMapping.shortName, com ManagementMapping.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime
cryptoService Mapping	CryptoServiceMapping	*	aggr	This aggregation represents the collection of crypto service mappings in the context of the enclosing System Mapping. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=cryptoServiceMapping.shortName, crypto ServiceMapping.variationPoint.shortLabel vh.latestBindingTime=postBuild
dataMapping	DataMapping	*	aggr	The data mappings defined. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=dataMapping, dataMapping.variation Point.shortLabel vh.latestBindingTime=postBuild
ecuResource Mapping	ECUMapping	*	aggr	Mapping of hardware related topology elements onto their counterpart definitions in the ECU Resource Template. atpVariation: The ECU Resource type might be variable. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=ecuResourceMapping.shortName, ecu ResourceMapping.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime
j1939Controller ApplicationTo J1939NmNode Mapping	J1939Controller ApplicationToJ1939Nm NodeMapping	*	aggr	Mapping of a J1939ControllerApplication to a J1939Nm Node.
mapping Constraint	MappingConstraint	*	aggr	Constraints that limit the mapping freedom for the mapping of SW components to ECUs. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=mappingConstraint, mapping Constraint.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime
pncMapping	PncMapping	*	aggr	Mappings between Virtual Function Clusters and Partial Network Clusters. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=pncMapping, pncMapping.variation Point.shortLabel vh.latestBindingTime=systemDesignTime
portElementTo ComResource Mapping	PortElementTo Communication ResourceMapping	*	aggr	maps a communication resource to CP Software Clusters **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=portElementToComResourceMapping.short Name, portElementToComResourceMapping.variation Point.shortLabel atp.Status=draft vh.latestBindingTime=postBuild

▽

△

Class	**SystemMapping**			
resource Estimation	EcuResourceEstimation	*	aggr	Resource estimations for this set of mappings, zero or one per ECU instance. atpVariation: Used ECUs are variable. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=resourceEstimation, resource Estimation.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime
resourceTo Application Partition Mapping	CpSoftwareCluster ResourceToApplication PartitionMapping	*	aggr	Maps a Software Cluster resource to an Application Partition to restrict the usage. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=resourceToApplicationPartition Mapping.shortName, resourceToApplicationPartition Mapping.variationPoint.shortLabel atp.Status=draft vh.latestBindingTime=systemDesignTime
rteEvent Separation	RteEventInSystem Separation	*	aggr	Separation constraint that limits the mapping freedom for the mapping of RteEvents to OsTasks in the System context.
rteEventToOs TaskProxy Mapping	RteEventInSystemToOs TaskProxyMapping	*	aggr	Constraint that enforces a mapping of RteEvent to a particular OsTask in the System context.
signalPath Constraint	SignalPathConstraint	*	aggr	Constraints that limit the mapping freedom for the mapping of data elements to signals. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=signalPathConstraint, signalPath Constraint.variationPoint.shortLabel vh.latestBindingTime=systemDesignTime
softwareCluster ToResource Mapping	CpSoftwareClusterTo ResourceMapping	*	aggr	maps a service resource to CP Software Clusters **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=softwareClusterToResourceMapping.short Name, softwareClusterToResourceMapping.variation Point.shortLabel atp.Status=draft vh.latestBindingTime=preCompileTime
swCluster Mapping	CpSoftwareClusterTo EcuInstanceMapping	*	aggr	The mappings of SW cluster to ECUs. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=swClusterMapping.shortName, swCluster Mapping.variationPoint.shortLabel atp.Status=draft vh.latestBindingTime=systemDesignTime
swcTo Application Partition Mapping	SwcToApplication PartitionMapping	*	aggr	Allows to map a given SwComponentPrototype to a formally defined partition at a point in time when the corresponding EcuInstance is not yet known or defined. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=swcToApplicationPartitionMapping.short Name, swcToApplicationPartitionMapping.variation Point.shortLabel vh.latestBindingTime=postBuild

▽

△

Class	SystemMapping			
swImplMapping	SwcToImplMapping	*	aggr	The mappings of AtomicSoftwareComponent Instances to Implementations. atpVariation: Derived, because SwcToEcuMapping is variable. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=swImplMapping.shortName, swImpl Mapping.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
swMapping	SwcToEcuMapping	*	aggr	The mappings of SW components to ECUs. atpVariation: SWC shall be mapped to other ECUs. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=swMapping.shortName, swMapping.variation Point.shortLabel vh.latestBindingTime=preCompileTime

Table A.868: SystemMapping

Class	SystemSignal			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	The system signal represents the communication system's view of data exchanged between SW components which reside on different ECUs. The system signals allow to represent this communication in a flattened structure, with exactly one system signal defined for each data element prototype sent and received by connected SW component instances. **Tags:**atp.recommendedPackage=SystemSignals			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dynamicLength	Boolean	1	attr	The length of dynamic length signals is variable in run-time. Only a maximum length of such a signal is specified in the configuration (attribute length in ISignal element).
physicalProps	SwDataDefProps	0..1	aggr	Specification of the physical representation. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=physicalProps

Table A.869: SystemSignal

Class	SystemSignalGroup	
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication	
Note	A signal group refers to a set of signals that shall always be kept together. A signal group is used to guarantee the atomic transfer of AUTOSAR composite data types. The SystemSignalGroup defines a signal grouping on VFB level. On cluster level the Signal grouping is described by the ISignalGroup element. **Tags:**atp.recommendedPackage=SystemSignalGroups	
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*	
Aggregated by	ARPackage.element	

▽

△

Class	SystemSignalGroup			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
systemSignal	SystemSignal	*	ref	Reference to a set of SystemSignals that shall always be kept together.
transforming SystemSignal	SystemSignal	0..1	ref	Optional reference to the SystemSignal which shall contain the transformed (linear) data.

Table A.870: SystemSignalGroup

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, TDEventSwcInternalBehavior and TDEventCom. **Tags:**atp.recommendedPackage=TimingExtensions			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*, *TimingExtension*			
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 A.871: SystemTiming

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. **Tags:**atp.Status=draft			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *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. **Tags:**atp.Status=draft
requestor	CpSoftwareCluster	*	ref	This is the software cluster that requests the temporal and dynamic resource. **Tags:**atp.Status=draft
timing Description	TimingDescription	0..1	ref	The timing description representing the temporal and dynamic resource. **Tags:**atp.Status=draft

Table A.872: TDCpSoftwareClusterMapping

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. **Tags:**atp.Status=draft			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
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. **Tags:**atp.Status=draft
timing Description	TimingDescription	0..1	ref	The timing description representing the temporal and dynamic resource. **Tags:**atp.Status=draft

Table A.873: TDCpSoftwareClusterResourceMapping

Class	TDEventBswInternalBehavior			
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescription Events::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**
bswModule Entity	BswModuleEntity	0..1	ref	The scope of this timing event.
tdEventBsw Internal BehaviorType	TDEventBswInternal BehaviorTypeEnum	0..1	attr	The specific type of this timing event.

Table A.874: TDEventBswInternalBehavior

Class	TDEventBswModeDeclaration			
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescription Events::TDEventBsw			
Note	This is used to describe timing events related to the mode communication on BSW level.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TDEventBsw*, *TimingDescription*, *Timing DescriptionEvent*			
Aggregated by	*TimingExtension*.timingDescription			
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	TDEventBswMode DeclarationTypeEnum	0..1	attr	The specific type of this timing event.

Table A.875: TDEventBswModeDeclaration

Class	TDEventBswModule			
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescription Events::TDEventBsw			
Note	This is used to describe timing events related to the interaction between BSW modules.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, TDEventBsw, TimingDescription, *Timing DescriptionEvent*			
Aggregated by	*TimingExtension*.timingDescription			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
bswModule Entry	BswModuleEntry	0..1	ref	The scope of this timing event.
tdEventBsw ModuleType	TDEventBswModule TypeEnum	0..1	attr	The specific type of this timing event.

Table A.876: TDEventBswModule

Class	TDEventComplex			
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescription Events::TDEventComplex			
Note	This is used to describe complex timing events. 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.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, TimingDescription, *TimingDescriptionEvent*			
Aggregated by	*TimingExtension*.timingDescription			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
–	–	–	–	–

Table A.877: TDEventComplex

Class	TDEventCycleStart (abstract)			
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescription Events::TDEventCom			
Note	This is the abstract parent class to describe timing events related to a point in time where a communication cycle starts. Via the attribute "cycleRepetition", a filtered view to the cycle start can be defined.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, TDEventCom, TimingDescription, *Timing DescriptionEvent*			
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 A.878: TDEventCycleStart

Class	TDEventFrClusterCycleStart	
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescription Events::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.	

▽

△

Class	TDEventFrClusterCycleStart			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, TDEventCom, *TDEventCycleStart*, *Timing Description*, *TimingDescriptionEvent*			
Aggregated by	*TimingExtension*.timingDescription			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
frCluster	FlexrayCluster	0..1	ref	The scope of this timing event.

Table A.879: TDEventFrClusterCycleStart

Class	TDEventFrame			
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescription Events::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*, *Timing DescriptionEvent*			
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 A.880: TDEventFrame

Class	TDEventFrameEthernet			
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescription Events::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*, *Timing DescriptionEvent*			
Aggregated by	*TimingExtension*.timingDescription			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
staticSocket Connection	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	TDEventFrameEthernet TypeEnum	0..1	attr	This is used to describe the specific event type of a 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 A.881: TDEventFrameEthernet

Class	TDEventIPdu			
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescription Events::TDEventCom			
Note	This is used to describe timing events related to the exchange of I-PDUs between the bus specific (Flex Ray / CAN / LIN) Interface BSW module and COM.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TDEventCom*, *TimingDescription*, *Timing DescriptionEvent*			
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 A.882: TDEventIPdu

Class	TDEventISignal			
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescription Events::TDEventCom			
Note	This is used to describe timing events related to the exchange of I-Signals between COM and RTE.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TDEventCom*, *TimingDescription*, *Timing DescriptionEvent*			
Aggregated by	*TimingExtension*.timingDescription			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
iSignal	ISignal	0..1	ref	The scope of this timing event.
physical Channel	PhysicalChannel	0..1	ref	The PhysicalChannel on which the ISignal is transmitted.
tdEventType	TDEventISignalType Enum	0..1	attr	The specific type of this timing event.

Table A.883: TDEventISignal

Class	TDEventModeDeclaration			
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescription Events::TDEventVfb::ModeDeclaration			
Note	This is used to describe timing events related to mode switch communication at VFB level.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TDEventVfb*, *TDEventVfbPort*, *Timing Description*, *TimingDescriptionEvent*			
Aggregated by	*TimingExtension*.timingDescription			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
entryMode Declaration	ModeDeclaration	0..1	ref	Optional parameter which refines the scope of the TDEventModeDeclaration. 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 TDEventModeDeclaration. 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 referenced mode declaration group prototype.
tdEventMode DeclarationType	TDEventMode DeclarationTypeEnum	0..1	attr	The specific type of this timing event.

Table A.884: TDEventModeDeclaration

Class	TDEventOccurrenceExpression			
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescription Events::TDEventOccurrenceExpression			
Note	This is used to specify a filter on the occurrences of TimingDescriptionEvents by means of a TDEvent OccurrenceExpressionFormula. 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	AutosarOperation ArgumentInstance	*	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	TDEventOccurrence ExpressionFormula	0..1	aggr	This is the expression formula which is used to describe the occurrence expression.
mode	TimingModeInstance	*	aggr	An occurrence expression can reference an arbitrary number of TimingModeInstances in its expression. This association aggregates instance references to Mode Declaration which can be referenced in the expression.
variable	AutosarVariable Instance	*	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 A.885: TDEventOccurrenceExpression

Class	TDEventOperation			
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescription Events::TDEventVfb::Operation			
Note	This is used to describe timing events related to client-server communication at VFB level.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TDEventVfb*, *TDEventVfbPort*, *Timing Description*, *TimingDescriptionEvent*			
Aggregated by	*TimingExtension*.timingDescription			
Attribute	Type	Mult.	Kind	Note
operation	ClientServerOperation	0..1	ref	The referenced operation.
tdEvent OperationType	TDEventOperationType Enum	0..1	attr	The specific type of this timing event.

Table A.886: TDEventOperation

Class	TDEventSwc (abstract)			
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescription Events::TDEventSwcInternalBehavior			
Note	This is the abstract parent class to describe timing events at Software Component (SW-C) level.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TimingDescription*, *TimingDescriptionEvent*			
Subclasses	TDEventSwcInternalBehavior, TDEventSwcInternalBehaviorReference			
Aggregated by	*TimingExtension*.timingDescription			
Attribute	Type	Mult.	Kind	Note

▽

△

Class	TDEventSwc (abstract)			
component	SwComponent Prototype	0..1	iref	The context for the scope of this timing event. **InstanceRef implemented by:**ComponentInComposition InstanceRef

Table A.887: TDEventSwc

Class	TDEventSwcInternalBehavior			
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescription Events::TDEventSwcInternalBehavior			
Note	This is used to describe timing events related to the SwcInternalBehavior of an AtomicSwComponent Type.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable, TDEventSwc, TimingDescription, Timing DescriptionEvent			
Aggregated by	TimingExtension.timingDescription			
Attribute	Type	Mult.	Kind	Note
runnable	RunnableEntity	0..1	ref	The scope of this timing event.
tdEventSwc Internal BehaviorType	TDEventSwcInternal BehaviorTypeEnum	0..1	attr	The specific type of this timing event.
variableAccess	VariableAccess	0..1	ref	The scope of this timing event.

Table A.888: TDEventSwcInternalBehavior

Class	TDEventSwcInternalBehaviorReference			
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescription Events::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	ARObject, Identifiable, MultilanguageReferrable, Referrable, TDEventSwc, TimingDescription, Timing DescriptionEvent			
Aggregated by	TimingExtension.timingDescription			
Attribute	Type	Mult.	Kind	Note
referenced TDEventSwc	TDEventSwc	0..1	ref	The referenced timing description event.

Table A.889: TDEventSwcInternalBehaviorReference

Class	TDEventTTCanCycleStart			
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescription Events::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.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable, TDEventCom, TDEventCycleStart, Timing Description, TimingDescriptionEvent			
Aggregated by	TimingExtension.timingDescription			
Attribute	Type	Mult.	Kind	Note
ttCanCluster	TtcanCluster	0..1	ref	The scope of this timing event.

Table A.890: TDEventTTCanCycleStart

Class	TDEventTrigger			
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescription Events::TDEventVfb::Trigger			
Note	This is used to describe timing events related to triggers at VFB level.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TDEventVfb*, *TDEventVfbPort*, *Timing Description*, *TimingDescriptionEvent*			
Aggregated by	*TimingExtension*.timingDescription			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
tdEventTrigger Type	TDEventTriggerType Enum	0..1	attr	The specific type of this timing event.
trigger	Trigger	0..1	ref	The trigger which is provided (released) or required (activate) in the given context.

Table A.891: TDEventTrigger

Class	TDEventVariableDataPrototype			
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescription Events::TDEventVfb::VariableDataPrototype			
Note	This is used to describe timing events related to sender-receiver communication at VFB level.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TDEventVfb*, *TDEventVfbPort*, *Timing Description*, *TimingDescriptionEvent*			
Aggregated by	*TimingExtension*.timingDescription			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
dataElement	VariableDataPrototype	0..1	ref	The referenced VariableDataPrototype
tdEventVariable DataPrototype Type	TDEventVariableData PrototypeTypeEnum	0..1	attr	The specific type of this timing event.

Table A.892: TDEventVariableDataPrototype

Class	TDEventVfb (abstract)			
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescription Events::TDEventVfb			
Note	This is the abstract parent class to describe timing events at Virtual Functional Bus (VFB) level.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TimingDescription*, *TimingDescriptionEvent*			
Subclasses	*TDEventVfbPort*, TDEventVfbReference			
Aggregated by	*TimingExtension*.timingDescription			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
component	SwComponent Prototype	0..1	iref	The context for the scope of this timing event. **InstanceRef implemented by:**ComponentInComposition InstanceRef

Table A.893: TDEventVfb

Class	TDEventVfbPort (abstract)			
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescription Events::TDEventVfb			
Note	This is the abstract parent class to describe specific timing event types at Virtual Functional Bus (VFB) level.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TDEventVfb*, *TimingDescription*, *Timing DescriptionEvent*			

▽

△

Class	TDEventVfbPort (abstract)			
Subclasses	TDEventModeDeclaration, TDEventOperation, TDEventTrigger, TDEventVariableDataPrototype			
Aggregated by	TimingExtension.timingDescription			
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	The port scope of the timing event.
portPrototype Blueprint	PortPrototypeBlueprint	0..1	ref	The PortPrototypeBlueprint is the scope of the timing event.

Table A.894: TDEventVfbPort

Class	TDEventVfbReference			
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription::TimingDescription Events::TDEventVfb			
Note	This is used to reference timing description events related to the Virtual Functional Bus (VFB) view which are specified in other timing views.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable, TDEventVfb, TimingDescription, Timing DescriptionEvent			
Aggregated by	TimingExtension.timingDescription			
Attribute	Type	Mult.	Kind	Note
referenced TDEventVfb	TDEventVfb	0..1	ref	The referenced timing description event.

Table A.895: TDEventVfbReference

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
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 A.896: TDHeaderIdRange

Class	TargetIPduRef			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Multiplatform			
Note	Target destination of the referencing mapping.			
Base	ARObject			
Aggregated by	IPduMapping.targetIPdu			

▽

△

Class	TargetIPduRef			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
defaultValue	PduMappingDefault Value	0..1	aggr	If no I-Pdu has been received a default value will be distributed.
targetIPdu	PduTriggering	1	ref	IPdu Reference

Table A.897: TargetIPduRef

Class	Tcplplcmpv4Props			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	This meta-class specifies the configuration options for ICMPv4 (Internet Control Message Protocol).			
Base	ARObject			
Aggregated by	EthTcpIpIcmpProps.icmpV4Props			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
tcplplcmp V4EchoReply Enabled	Boolean	0..1	attr	This attribute enables or disables transmission of ICMP echo reply message in case of a ICMP echo reception.
tcplplcmpV4Ttl	PositiveInteger	0..1	attr	This attribute is only relevant in case that ICMP (Internet Control Message Protocol) is used. It specifies the default Time-to-live value of outgoing ICMP packets.

Table A.898: Tcplplcmpv4Props

Class	Tcplplcmpv6Props			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	This meta-class specifies the configuration options for ICMPv6 (Internet Control Message Protocol).			
Base	ARObject			
Aggregated by	EthTcpIpIcmpProps.icmpV6Props			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
tcplplcmp V6EchoReply Avoid Fragmentation	Boolean	0..1	attr	This attribute defines whether the echo reply is only transmitted in case that the incoming ICMPv6 Echo Request (Pings) fits the MTU of the respective interface, i.e. can be transmitted without IPv6 fragmentation.
tcplplcmp V6EchoReply Enabled	Boolean	0..1	attr	This attribute enables or disables transmission of ICMP echo reply message in case of a ICMP echo reception.
tcplplcmp V6HopLimit	PositiveInteger	0..1	attr	Default Hop-Limit value of outgoing ICMPv6 packets.
tcplplcmp V6Msg Destination Unreachable Enabled	Boolean	0..1	attr	This attribute Enables/Disables the transmission of Destination Unreachable Messages.
tcplplcmp V6Msg Parameter Problem Enabled	Boolean	0..1	attr	If enabled an ICMPv6 parameter problem message will be sent if a received packet has been dropped due to unknown options or headers that are found in the packet.

Table A.899: Tcplplcmpv6Props

Class	TcpProps			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	This meta-class specifies the configuration options for TCP (Transmission Control Protocol).			
Base	*ARObject*			
Aggregated by	EthTcpIpProps.tcpProps			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
tcpCongestion Avoidance Enabled	Boolean	0..1	attr	Enables (TRUE) or disables (FALSE) support of TCP congestion avoidance algorithm according to IETF RFC 5681.
tcpDelayedAck Timeout	TimeValue	0..1	attr	The maximal time an acknowledgement is delayed for transmission in seconds.
tcpFast Recovery Enabled	Boolean	0..1	attr	Enables (TRUE) or disables (FALSE) support of TCP Fast Recovery according to IETF RFC 5681.
tcpFast Retransmit Enabled	Boolean	0..1	attr	Enables (TRUE) or disables (FALSE) support of TCP Fast Retransmission according to IETF RFC 5681.
tcpFin Wait2Timeout	TimeValue	0..1	attr	Timeout in [s] to receive a FIN from the remote node (after this node has initiated connection termination), i.e. maximum time waiting in FINWAIT-2 for a connection termination request from the remote TCP.
tcpKeepAlive Enabled	Boolean	0..1	attr	Enables (TRUE) or disables (FALSE) TCP Keep Alive Probes according to IETF RFC 1122 chapter 4.2.3.6.
tcpKeepAlive Interval	TimeValue	0..1	attr	Specifies the interval in seconds between subsequent keepalive probes.
tcpKeepAlive ProbesMax	PositiveInteger	0..1	attr	Maximum number of times that a TCP Keep Alive is retransmitted before the connection is closed.
tcpKeepAlive Time	TimeValue	0..1	attr	Specifies the time in [s] between the last data packet sent (simple ACKs are not considered data) and the first keepalive probe.
tcpMaxRtx	PositiveInteger	0..1	attr	Maximum number of times that a TCP segment is retransmitted before the TCP connection is closed. This parameter is only valid if tcpRetransmissionTimeout is configured. Note: This parameter also applies for FIN retransmissions.
tcpMsl	TimeValue	0..1	attr	Maximum segment lifetime in [s].
tcpNagle Enabled	Boolean	0..1	attr	Enables (TRUE) or disables (FALSE) support of Nagle's algorithm according to IETF RFC 1122 (chapter 4.2.3.4 When to Send Data). If enabled the Nagle's algorithm is activated per default for all TCP sockets, but can be deactivated per Socket (with the attribute TcpTp.nagle Algorithm).
tcpReceive WindowMax	PositiveInteger	0..1	attr	Default value of maximum receive window in bytes.
tcp Retransmission Timeout	TimeValue	0..1	attr	Timeout in [s] before an unacknowledged TCP segment is sent again. If the timeout is disabled, no TCP segments shall be retransmitted.
tcpSlowStart Enabled	Boolean	0..1	attr	Enables (TRUE) or disables (FALSE) support of TCP slow start algorithm according to IETF RFC 5681.
tcpSynMaxRtx	PositiveInteger	0..1	attr	Maximum number of times that a TCP SYN is retransmitted.
tcpSynReceived Timeout	TimeValue	0..1	attr	Timeout in [s] to complete a remotely initiated TCP connection establishment, i.e. maximum time waiting in SYN-RECEIVED for a confirming connection request acknowledgement after having both received and sent a connection request.
tcpTtl	PositiveInteger	0..1	attr	Default Time-to-live value of outgoing TCP packets.

Table A.900: TcpProps

Class	TcpTp			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Content Model for TCP configuration.			
Base	*ARObject*, *TcpUdpConfig*, *TransportProtocolConfiguration*			
Aggregated by	ApApplicationEndpoint.tpConfiguration, ApplicationEndpoint.tpConfiguration, HttpTp.tcpTpConfig, RtpTp.tcpUdpConfig			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
keepAlive Interval	TimeValue	0..1	attr	Specifies the interval in seconds between subsequent keepalive probes.
keepAlive ProbesMax	PositiveInteger	0..1	attr	Maximum number of times that TCP retransmits an individual data segment before aborting the connection.
keepAlives	Boolean	0..1	attr	Indicates if Keep-Alive messages are sent.
keepAliveTime	TimeValue	0..1	attr	Specifies the time in seconds between the last data packet sent and the first keepalive probe.
naglesAlgorithm	Boolean	0..1	attr	Indicates if Nagle's Algorithm is used.
receiveWindow Min	PositiveInteger	0..1	attr	Minimum size of the TCP receive window in bytes.
tcp Retransmission Timeout	TimeValue	0..1	attr	Defines the timeout in seconds before an unacknowledged TCP segment is sent again. If the tcp RetransmissionTimeout is not defined or set to "INF", no TCP segments shall be re-transmitted.
tcpTpPort	TpPort	1	aggr	TCP Port configuration.

Table A.901: TcpTp

Class	TextTableMapping			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	Defines the mapping of two DataPrototypes typed by AutosarDataTypes that refer to CompuMethods of category TEXTTABLE, SCALE_LINEAR_AND_TEXTTABLE or BITFIELD_TEXTTABLE.			
Base	*ARObject*			
Aggregated by	DataPrototypeMapping.textTableMapping, SenderRecArrayTypeMapping.senderToSignalTextTable Mapping, SenderRecArrayTypeMapping.signalToReceiverTextTableMapping, SenderReceiverToSignal Mapping.senderToSignalTextTableMapping, SenderReceiverToSignalMapping.signalToReceiverTextTable Mapping, SenderRecRecordElementMapping.senderToSignalTextTableMapping, SenderRecRecord ElementMapping.signalToReceiverTextTableMapping, SubElementMapping.textTableMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
bitfieldTextTable MaskFirst	PositiveInteger	0..1	attr	This attribute can be used to support the mapping of bit field to bit field, boolean values to bit fields, and vice versa. The attribute defines the bit mask for the first element of the TextTableMapping. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime
bitfieldTextTable MaskSecond	PositiveInteger	0..1	attr	This attribute can be used to support the mapping of bit field to bit field, boolean values to bit fields, and vice versa. The attribute defines the bit mask for the second element of the TextTableMapping. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime
identical Mapping	Boolean	0..1	attr	If identicalMapping is set == true the values of the two referenced DataPrototypes do not need any conversion of the values.
mapping Direction	MappingDirectionEnum	0..1	attr	Specifies the conversion direction for which the TextTable Mapping is applicable.
valuePair	TextTableValuePair	*	aggr	Defines a pair of values which are translated into each other.

Table A.902: TextTableMapping

Class	TextTableValuePair			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	Defines a pair of text values which are translated into each other.			
Base	ARObject			
Aggregated by	TextTableMapping.valuePair			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
firstValue	Numerical	0..1	attr	Value of first DataPrototype provided similar to a numerical ValueSpecification which is intended to be assigned to a Primitive data element. Note that the numerical value is a variant, it can be computed by a formula. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime
secondValue	Numerical	0..1	attr	Value of second DataPrototype provided similar to a numerical ValueSpecification which is intended to be assigned to a Primitive data element. Note that the numerical value is a variant, it can be computed by a formula. **Stereotypes:** atpVariation **Tags:**vh.latestBindingTime=preCompileTime

Table A.903: TextTableValuePair

Class	TextValueSpecification			
Package	M2::AUTOSARTemplates::CommonStructure::Constants			
Note	The purpose of TextValueSpecification is to define the labels that correspond to enumeration values.			
Base	ARObject, *ValueSpecification*			
Aggregated by	ApplicationAssocMapElementValueSpecification.key, ApplicationAssocMapElementValueSpecification. value, ArrayValueSpecification.element, CalibrationParameterValue.applInitValue, CalibrationParameter Value.implInitValue, ConstantSpecification.valueSpec, CryptoServiceKey.developmentValue, Diagnostic EnvDataCondition.compareValue, DiagnosticEnvDataElementCondition.compareValue, FieldSenderCom Spec.initValue, ISignal.initValue, ISignal.timeoutSubstitutionValue, MetaDataItem.metaDataItemType, NonqueuedReceiverComSpec.initValue, NonqueuedReceiverComSpec.timeoutSubstitutionValue, NonqueuedSenderComSpec.initValue, NvProvideComSpec.ramBlockInitValue, NvProvideComSpec.rom BlockInitValue, NvRequireComSpec.initValue, ParameterDataPrototype.initValue, ParameterProvideCom Spec.initValue, ParameterRequireComSpec.initValue, PersistencyDataRequiredComSpec.initValue, PersistencyKeyValuePair.initValue, PortDefinedArgumentValue.value, PortPrototypeBlueprintInitValue. value, RecordValueSpecification.field, *StateManagementCompareCondition*.compareValue, SwDataDef Props.invalidValue, VariableDataPrototype.initValue			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
value	VerbatimString	0..1	attr	This is the value itself. Note that vt uses the | operator to separate the values for the different bitfield masks in case that the semantics of the related DataPrototype is described by means of a BITFIELD_TEXTTABLE in the associated CompuMethod.

Table A.904: TextValueSpecification

Class	TimeSyncClientConfiguration			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Defines the configuration of the time synchronisation client.			
Base	ARObject			
Aggregated by	TimeSynchronization.timeSyncClient			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	TimeSyncClientConfiguration			
orderedMaster (ordered)	OrderedMaster	*	aggr	Defines a list of ordered NetworkEndpoints. **Tags:**xml.namePlural=ORDERED-MASTER-LIST
timeSync Technology	TimeSyncTechnology Enum	1	attr	Defines the time synchronisation technology used.

Table A.905: TimeSyncClientConfiguration

Class	TimeSyncServerConfiguration			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Defines the configuration of the time synchronisation server.			
Base	ARObject, *Referrable*			
Aggregated by	TimeSynchronization.timeSyncServer			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
priority	PositiveInteger	0..1	attr	Server Priority.
syncInterval	TimeValue	1	attr	Synchronisation interval used by the time synchronisation server (in seconds).
timeSyncServer Identifier	String	0..1	attr	Identifier of the TimeSyncServer.
timeSync Technology	TimeSyncTechnology Enum	1	attr	Defines the time synchronisation technology used. Possible values are: NTP_RFC958, PTP_ IEEE1588_2002, PTP_IEEE1588_2008, AVB_ IEEE802_1AS and others.

Table A.906: TimeSyncServerConfiguration

Class	*TimingDescriptionEvent* (abstract)			
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingDescription			
Note	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. In order to avoid confusion with existing event descriptions in the AUTOSAR templates the timing specific event types use the prefix TD.			
Base	ARObject, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TimingDescription*			
Subclasses	*TDEventBsw*, *TDEventBswInternalBehavior*, *TDEventCom*, *TDEventComplex*, *TDEventSLLET*, *TDEvent Swc*, *TDEventVfb*			
Aggregated by	*TimingExtension*.timingDescription			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
clockReference	TimingClock	0..1	ref	Optional reference to a clock that holds the time base for an TD event. **Tags:**atp.Status=draft
occurrence Expression	TDEventOccurrence Expression	0..1	aggr	The occurrence expression for this event.

Table A.907: TimingDescriptionEvent

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 *event chain segments*.
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 A.908: TimingDescriptionEventChain

Class	TimingEvent
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents
Note	This event is used to start RunnableEntities that shall be executed periodically.
Base	*ARObject*, *AbstractEvent*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *Multilanguage Referrable*, *RTEEvent*, *Referrable*
Aggregated by	*AtpClassifier*.atpFeature, SwcInternalBehavior.event

Attribute	Type	Mult.	Kind	Note
offset	TimeValue	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	TimeValue	0..1	attr	Period of timing event in seconds. The value of this attribute shall be greater than zero.

Table A.909: TimingEvent

Class	*TimingExtension* (abstract)
Package	M2::AUTOSARTemplates::CommonStructure::Timing::TimingExtensions
Note	The abstract parent class of the different template specific timing extensions. Depending on the specific timing extension the timing descriptions and timing constraints, that can be used to specify the timing behavior, are restricted.
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*
Subclasses	BswCompositionTiming, BswModuleTiming, EcuTiming, SwcTiming, SystemTiming, VfbTiming

▽

Document ID 635: AUTOSAR_TR_ClassicAutosarModelConstraints

△

Class	**TimingExtension** (abstract)			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
timingClock	TimingClock	*	aggr	A list of abstract model Clocks. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=timingClock.shortName, timing Clock.variationPoint.shortLabel atp.Status=draft vh.latestBindingTime=postBuild
timingClock SyncAccuracy	TimingClockSync Accuracy	*	aggr	A list of accuracies - which may be used to specify synchronizations from one model clock to another model clock. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=timingClockSyncAccuracy.shortName, timing ClockSyncAccuracy.variationPoint.shortLabel atp.Status=draft vh.latestBindingTime=postBuild
timingCondition	TimingCondition	*	aggr	The timing condition specifies a specific condition. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=timingCondition.shortName, timing Condition.variationPoint.shortLabel vh.latestBindingTime=postBuild
timing Description	TimingDescription	*	aggr	The timing descriptions that belong to a specific timing specification. In order to support different timing description variants within a timing specification, the aggregation is marked with the stereotype "atpVariation". **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=timingDescription.shortName, timing Description.variationPoint.shortLabel vh.latestBindingTime=postBuild
timing Guarantee	TimingConstraint	*	aggr	The timing constraints that belong to a specific timing specification in the role of a timing guarantee. In order to support different timing constraint variants within a timing specification, the aggregation is marked with the stereotype "atpVariation". **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=timingGuarantee.shortName, timing Guarantee.variationPoint.shortLabel vh.latestBindingTime=postBuild
timing Requirement	TimingConstraint	*	aggr	The timing constraints that belong to a specific timing specification in the role of a timing requirement. In order to support different timing constraint variants within a timing specification, the aggregation is marked with the stereotype "atpVariation". **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=timingRequirement.shortName, timing Requirement.variationPoint.shortLabel vh.latestBindingTime=postBuild

▽

△

Class	TimingExtension (abstract)			
timingResource	TimingExtension Resource	0..1	aggr	The timing resource contains all instance references referred from within a timing condition formula of a timing view. **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=timingResource.shortName

Table A.910: TimingExtension

Class	TlsCryptoCipherSuite			
Package	M2::AUTOSARTemplates::SystemTemplate::SecureCommunication			
Note	This meta-class represents a cipher suite for describing cryptographic operations in the context of establishing a connection of ApplicationEndpoints that is protected by TLS.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	TlsCryptoServiceMapping.tlsCipherSuite, TlsSecureComProps.tlsCipherSuite			
Attribute	Type	Mult.	Kind	Note
authentication	CryptoServicePrimitive	0..1	ref	This reference identifies the crypto service primitive for the generation and verification of MACs.
certificate	CryptoServiceCertificate	0..1	ref	This reference identifies the applicable local certificate.
cipherSuiteId	PositiveInteger	0..1	attr	Identification of the CipherSuite according to the IANA assignments list.
cipherSuiteShortLabel	String	0..1	attr	Name of the CipherSuite according to the IANA assignments list.
ellipticCurve	CryptoEllipticCurveProps	*	ref	This references point to the properties of elliptic curves.
encryption	CryptoServicePrimitive	0..1	ref	This reference identifies the crypto service primitive for the execution of encryption.
keyExchange	CryptoServicePrimitive	*	ref	This reference identifies the individual (i.e. per cipher suite) crypto service primitive for the execution of key exchange during the handshake phase.
keyExchangeAuthentication	CryptoServicePrimitive	*	ref	This reference identifies the crypto service primitives for the generation and verification of signatures during the key exchange algorithm.
priority	PositiveInteger	0..1	attr	This attribute identifies the priority of the cipher suite. Range: 1..65535. Lower values represent higher priorities.
props	TlsCryptoCipherSuiteProps	0..1	aggr	The aggregated TlsCryptoCipherSuiteProps provide details for the TLS Cipher Suite.
pskIdentity	TlsPskIdentity	0..1	aggr	Pre-shared key identity shared during the handshake among the communication parties, to establish a TLS connection if the handshake is based on the existence of a pre-shared key.
remoteCertificate	CryptoServiceCertificate	0..1	ref	This reference identifies the applicable remote certificate.
signatureScheme	CryptoSignatureScheme	*	ref	This reference points to the properties of a TLS Signature Scheme.
version	TlsVersionEnum	1	attr	This attribute supports the definition of the applicable version of TLS.

Table A.911: TlsCryptoCipherSuite

Class	TlsCryptoServiceMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::SecureCommunication			
Note	This meta-class has the ability to represent a crypto service mapping for the socket-based configuration of Transport Layer Security (TLS).			
Base	*ARObject*, *CryptoServiceMapping*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	SystemMapping.cryptoServiceMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
keyExchange	CryptoServicePrimitive	*	ref	This reference identifies the shared(i.e. applicable for each of the aggregated cipher suites) crypto service primitive for the execution of key exchange during the handshake phase.
tlsCipherSuite	TlsCryptoCipherSuite	*	aggr	This aggregation represents the collection of supported cipher suites.
useClient Authentication Request	Boolean	0..1	attr	Defines if client authentication shall be applied for this TLS connection.
useSecurity Extension RecordSize Limit	Boolean	0..1	attr	Defines if the security extension for max_fragment_length shall be supported as defined in IETF RFC 8449, chapter 4.1.

Table A.912: TlsCryptoServiceMapping

Class	TlsPskIdentity			
Package	M2::AUTOSARTemplates::SystemTemplate::SecureCommunication			
Note	This element is used to describe the pre-shared key shared during the handshake among the communication parties, to establish a TLS connection if the handshake is based on the existence of a pre-shared key.			
Base	*ARObject*			
Aggregated by	TlsCryptoCipherSuite.pskIdentity			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
preSharedKey	CryptoServiceKey	1	ref	This reference identifies the applicable cryptographic key.
pskIdentity	String	1	attr	This attribute provides the key identification.
pskIdentityHint	String	0..1	attr	This attribute provides the identity hint for a pre-shared key.

Table A.913: TlsPskIdentity

Class	TlvDataIdDefinition			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	This meta-class represents the ability to define the tlvDataId.			
Base	*ARObject*			
Aggregated by	TlvDataIdDefinitionSet.tlvDataIdDefinition			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
id	PositiveInteger	1	attr	This attribute represents the definition of the value of the TlvDataId **Stereotypes:** atpIdentityContributor
tlvArgument	ArgumentDataPrototype	0..1	ref	This reference assigns a tlvDataId to a given argument of a ClientServerOperation.

▽

△

Class	TlvDataIdDefinition			
tlv Implementation DataType Element	AbstractImplementation DataTypeElement	0..1	ref	This reference associates the definition of a TLV data id with a given AbstractImplementationDataTypeElement.
tlvRecord Element	ApplicationRecord Element	0..1	ref	This reference associates the definition of a TLV data id with a given ApplicationRecordElement.

Table A.914: TlvDataIdDefinition

Class	TlvDataIdDefinitionSet			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	This meta-class acts as a container of TlvDataIdDefinitions to be used in a given context **Tags:**atp.recommendedPackage=TlvDataDefinitionSets			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
tlvDataId Definition	TlvDataIdDefinition	*	aggr	This aggregation represents the collection of TlVDataTid Definitions aggregated by the TlvDataIdDefinitionSet **Stereotypes:** atpSplitable **Tags:**atp.Splitkey=tlvDataIdDefinition.id

Table A.915: TlvDataIdDefinitionSet

Class	Topic1			
Package	M2::MSR::Documentation::Chapters			
Note	This meta-class represents a topic of a documentation. Topics are similar to chapters but they cannot be nested. They also do not appear in the table of content. Topics can be used to produce intermediate headlines thus structuring a chapter internally.			
Base	*ARObject*, *DocumentViewSelectable*, *Identifiable*, *MultilanguageReferrable*, *Paginateable*, *Referrable*			
Aggregated by	MsrQueryResultTopic1.topic1, TopicOrMsrQuery.topic1			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
helpEntry	String	0..1	attr	This specifies an entry point in an online help system to be linked with the parent class. The syntax shall be defined by the applied help system respectively help system generator. **Tags:**xml.attribute=true
topicContent	TopicContentOrMsr Query	0..1	aggr	This is the content of the topic. **Tags:** xml.roleElement=false xml.roleWrapperElement=false xml.sequenceOffset=20 xml.typeElement=false xml.typeWrapperElement=false

Table A.916: Topic1

Class	**TpConnection** (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::DiagnosticConnection			
Note	TpConnection Base Class.			
Base	ARObject			
Subclasses	CanTpConnection, DoIpTpConnection, EthTpConnection, FlexrayArTpConnection, FlexrayTpConnection, J1939TpConnection, LinTpConnection			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
ident	TpConnectionIdent	0..1	aggr	This adds the ability to become referrable to Tp Connection.

Table A.917: TpConnection

Class	**TpConnectionIdent**			
Package	M2::AUTOSARTemplates::SystemTemplate::DiagnosticConnection			
Note	This meta-class is created to add the ability to become the target of a reference to the non-Referrable Tp Connection.			
Base	ARObject, Referrable			
Aggregated by	TpConnection.ident			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
–	–	–	–	–

Table A.918: TpConnectionIdent

Class	**TpPort**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Dynamic or direct assignment of a PortNumber.			
Base	ARObject			
Aggregated by	TcpTp.tcpTpPort, UdpTp.udpTpPort			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dynamically Assigned	Boolean	0..1	attr	Indicates whether the source port is dynamically assigned. **Tags:**atp.Status=obsolete
portNumber	PositiveInteger	0..1	attr	Port Number.

Table A.919: TpPort

Class	**Traceable** (abstract)			
Package	M2::MSR::Documentation::BlockElements::RequirementsTracing			
Note	This meta class represents the ability to be subject to tracing within an AUTOSAR model. Note that it is expected that its subclasses inherit either from MultilanguageReferrable or from Identifiable. Nevertheless it also inherits from MultilanguageReferrable in order to provide a common reference target for all Traceables.			
Base	ARObject, MultilanguageReferrable, Referrable			
Subclasses	StructuredReq, TimingConstraint, TraceableTable, TraceableText			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	Traceable (abstract)			
trace	Traceable	*	ref	This association represents the ability to trace to upstream requirements / constraints. This supports for example the bottom up tracing
				ProjectObjectives <- MainRequirements <- Features <- RequirementSpecs <- BSW/AI
				Tags:xml.sequenceOffset=20

Table A.920: Traceable

Class	TraceableText			
Package	M2::MSR::Documentation::BlockElements::RequirementsTracing			
Note	This meta-class represents the ability to denote a traceable text item such as requirements etc. The following approach applies: • **shortName** represents the tag for tracing • **longName** represents the head line • **category** represents the kind of the tagged text			
Base	*ARObject*, *DocumentViewSelectable*, *Identifiable*, *MultilanguageReferrable*, *Paginateable*, *Referrable*, *Traceable*			
Aggregated by	DocumentationBlock.trace			
Attribute	Type	Mult.	Kind	Note
text	DocumentationBlock	1	aggr	This represents the text to which the tag applies. **Tags:** xml.roleElement=false xml.roleWrapperElement=false xml.sequenceOffset=30 xml.typeElement=false xml.typeWrapperElement=false

Table A.921: TraceableText

Enumeration	TransferPropertyEnum	
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication	
Note	Transfer Properties of a Signal.	
Aggregated by	ISignalToIPduMapping.transferProperty	
Literal	Description	
pending	If the signal has the TransferProperty pending, then the function Com_SendSignal shall not perform a transmission of the IPdu associated with the signal. **Tags:**atp.EnumerationLiteralIndex=0	
triggered	The signal in the assigned IPdu is updated and a request for the IPdu's transmission is made. **Tags:**atp.EnumerationLiteralIndex=1	
triggeredOnChange	The signal in the assigned IPdu is updated and a request for the IPdus transmission is made only if the signal value is different from the already stored signal value. **Tags:**atp.EnumerationLiteralIndex=2	
triggeredOnChange WithoutRepetition	The signal in the assigned IPdu is updated and a request for the IPdus transmission is made only if the signal value is different from the already stored signal value. In the DIRECT/N-TIMES or MIXED transmission mode (EventControlledTiming) the IPdu will be transmitted just once without a repetition, independent of the defined NumberOfRepeats. **Tags:**atp.EnumerationLiteralIndex=3	

▽

△

Enumeration	TransferPropertyEnum
triggeredWithout Repetition	The signal in the assigned IPdu is updated and a request for the IPdu's transmission is made. In the DIRECT/N-TIMES or MIXED transmission mode (EventControlledTiming) the IPdu will be transmitted just once without a repetition, independent of the defined NumberOfRepeats. **Tags:**atp.EnumerationLiteralIndex=4

Table A.922: TransferPropertyEnum

Class	*TransformationDescription* (abstract)
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer
Note	The TransformationDescription is the abstract class that can be used by specific transformers to add transformer specific properties.
Base	*ARObject*, *Describable*
Subclasses	EndToEndTransformationDescription, SOMEIPTransformationDescription, UserDefinedTransformation Description
Aggregated by	TransformationTechnology.transformationDescription
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.923: TransformationDescription

Class	<<atpVariation>> *TransformationISignalProps* (abstract)
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer
Note	TransformationISignalProps holds all the attributes for the different TransformationTechnologies that are ISignal specific. **Tags:**vh.latestBindingTime=postBuild
Base	*ARObject*, *Describable*
Subclasses	EndToEndTransformationISignalProps, SOMEIPTransformationISignalProps, UserDefinedTransformation ISignalProps
Aggregated by	ISignal.transformationISignalProps, ISignalGroup.transformationISignalProps

Attribute	Type	Mult.	Kind	Note
csErrorReaction	CSTransformerError ReactionEnum	0..1	attr	Defines whether the transformer chain of client/server communication coordinates an autonomous error reaction together with the RTE or whether any error reaction is the responsibility of the application.
dataPrototype Transformation Props	DataPrototype TransformationProps	*	aggr	Fine granular modeling of TransfromationProps on the level of DataPrototypes.
transformer	Transformation Technology	1	ref	Reference to the TransformationTechnology description that contains transformer specific and ISignal independent configuration properties.

Table A.924: TransformationISignalProps

Class	TransformationTechnology
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer
Note	A TransformationTechnology is a transformer inside a transformer chain. **Tags:**xml.namePlural=TRANSFORMATION-TECHNOLOGIES
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*

▽

△

Class	TransformationTechnology			
Aggregated by	DataTransformationSet.transformationTechnology			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
bufferProperties	BufferProperties	1	aggr	Aggregation of the mandatory BufferProperties.
hasInternal State	Boolean	0..1	attr	This attribute defines whether the Transformer has an internal state or not.
needsOriginal Data	Boolean	0..1	attr	Specifies whether this transformer gets access to the SWC's original data.
protocol	String	1	attr	Specifies the protocol that is implemented by this transformer.
transformation Description	Transformation Description	0..1	aggr	A transformer can be configured with transformer specific parameters which are represented by the Transformer Description. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=transformationDescription, transformation Description.variationPoint.shortLabel vh.latestBindingTime=postBuild
transformer Class	TransformerClassEnum	1	attr	Specifies to which transformer class this transformer belongs.
version	String	1	attr	Version of the implemented protocol.

Table A.925: TransformationTechnology

Enumeration	TransformerClassEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer
Note	Specifies the transformer class of a transformer.
Aggregated by	TransformationTechnology.transformerClass
Literal	**Description**
custom	The transformer is a custom transformer. **Tags:**atp.EnumerationLiteralIndex=0
safety	The transformer is a safety transformer. **Tags:**atp.EnumerationLiteralIndex=1
security	The transformer is a security transformer. **Tags:**atp.EnumerationLiteralIndex=2
serializer	The transformer is a serializing transformer. **Tags:**atp.EnumerationLiteralIndex=3

Table A.926: TransformerClassEnum

Class	TransformerHardErrorEvent			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents			
Note	This event is raised when data are received which should trigger a Client/Server operation or an external Trigger but during transformation of the data a hard transformer error occurred.			
Base	*ARObject*, *AbstractEvent*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *Multilanguage Referrable*, *RTEEvent*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, SwcInternalBehavior.event			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	TransformerHardErrorEvent			
operation	ClientServerOperation	0..1	iref	This represents the ClientServerOperation for which the transformer can raise this TransformerHardErrorEvent. **InstanceRef implemented by:**POperationInAtomicSwcInstanceRef
requiredTrigger	Trigger	0..1	iref	This represents the Trigger for which the transformer can raise this TransformerHardErrorEvent. **InstanceRef implemented by:**RTriggerInAtomicSwcInstanceRef

Table A.927: TransformerHardErrorEvent

Class	TransientFault			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	The reported failure is classified as runtime error.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable, TracedFailure			
Aggregated by	ErrorTracerNeeds.tracedFailure			
Attribute	Type	Mult.	Kind	Note
possibleErrorReaction	PossibleErrorReaction	*	aggr	Describes a possible error reactions for the transient fault handler.

Table A.928: TransientFault

Class	TransmissionAcknowledgementRequest			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	Requests transmission acknowledgement that data has been sent successfully. Success/failure is reported via a SendPoint of a RunnableEntity.			
Base	ARObject			
Aggregated by	SenderComSpec.transmissionAcknowledge			
Attribute	Type	Mult.	Kind	Note
timeout	TimeValue	0..1	attr	Number of seconds before an error is reported or in case of allowed redundancy, the value is sent again.

Table A.929: TransmissionAcknowledgementRequest

Class	TransmissionComSpecProps			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	This meta-class defines a set of transmission attributes which the application software is assumed to implement.			
Base	ARObject			
Aggregated by	SenderComSpec.transmissionProps			
Attribute	Type	Mult.	Kind	Note
dataUpdatePeriod	TimeValue	0..1	attr	This attribute defines the period in which the application is assumed to transmit the respective data.
minimumSendInterval	TimeValue	0..1	attr	This attribute defines the minimum interval between two consecutive transmissions of the respective data the application is assumed to ensure.
transmissionMode	TransmissionModeDefinitionEnum	0..1	attr	The attribute defines the mode in which the application is assumed to transmit the respective data.

Table A.930: TransmissionComSpecProps

Class	TransmissionModeCondition			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing			
Note	Possibility to attach a condition to each signal within an I-PDU.			
	If at least one condition evaluates to true, TRANSMISSION MODE True shall be used for this I-Pdu. In all other cases, the TRANSMISSION MODE FALSE shall be used.			
Base	ARObject			
Aggregated by	TransmissionModeDeclaration.transmissionModeCondition			
Attribute	Type	Mult.	Kind	Note
dataFilter	DataFilter	1	aggr	Possibilities to define conditions
iSignalInIPdu	ISignalToIPduMapping	1	ref	Reference to a signal to which a condition is attached.

Table A.931: TransmissionModeCondition

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, Identifiable, MultilanguageReferrable, Referrable			
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 A.932: Trigger

Class	TriggerInterface			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	A trigger interface declares a number of triggers that can be sent by an trigger source.			
	Tags:atp.recommendedPackage=PortInterfaces			
Base	ARElement, ARObject, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, PortInterface, Referrable			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
trigger	Trigger	*	aggr	The Trigger of this trigger interface.

Table A.933: TriggerInterface

Class	TriggerInterfaceMapping			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	Defines the mapping of unequal named Triggers in context of two different TriggerInterfaces.			
Base	ARObject, AtpBlueprint, AtpBlueprintable, Identifiable, MultilanguageReferrable, PortInterfaceMapping, Referrable			
Aggregated by	PortInterfaceMappingSet.portInterfaceMapping			
Attribute	Type	Mult.	Kind	Note
triggerMapping	TriggerMapping	*	aggr	Mapping of two Trigger in two different TriggerInterface

Table A.934: TriggerInterfaceMapping

Class	TriggerMapping			
Package	M2::AUTOSARTemplates::CommonStructure::TriggerDeclaration			
Note	Defines the mapping of two particular unequally named Triggers in the given context.			
Base	ARObject			
Aggregated by	TriggerInterfaceMapping.triggerMapping			
Attribute	Type	Mult.	Kind	Note
firstTrigger	Trigger	0..1	ref	A Trigger to be mapped.
secondTrigger	Trigger	0..1	ref	A Trigger to be mapped.

Table A.935: TriggerMapping

Class	TriggerPortAnnotation			
Package	M2::AUTOSARTemplates::SWComponentTemplate::ApplicationAttributes			
Note	Annotation to a port used for calibration regarding a certain Trigger.			
Base	ARObject, GeneralAnnotation			
Aggregated by	PortPrototype.triggerPortAnnotation			
Attribute	Type	Mult.	Kind	Note
trigger	Trigger	0..1	ref	The instance of annotated trigger.

Table A.936: TriggerPortAnnotation

Class	TriggerToSignalMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	This meta-class represents the ability to map a trigger to a SystemSignal of size 0. The Trigger does not transport any other information than its existence, therefore the limitation in terms of signal length.			
Base	ARObject, DataMapping			
Aggregated by	SystemMapping.dataMapping			
Attribute	Type	Mult.	Kind	Note
systemSignal	SystemSignal	1	ref	This is the SystemSignal taken to transport the Trigger over the network. **Tags:**xml.sequenceOffset=20
trigger	Trigger	1	iref	This represents the Trigger that shall be used to trigger RunnableEntities deployed to a remote ECU. **Tags:**xml.sequenceOffset=10 **InstanceRef implemented by:**TriggerInSystemInstanceRef

Table A.937: TriggerToSignalMapping

Class	<<atpVariation>> TtcanCluster			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ttcan::TtcanTopology			
Note	TTCAN bus specific cluster attributes. **Tags:**atp.recommendedPackage=CommunicationClusters			
Base	ARObject, AbstractCanCluster, CollectableElement, CommunicationCluster, FibexElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note

▽

△

Class	<<atpVariation>> **TtcanCluster**			
basicCycle Length	Integer	1	attr	Length of a basic-cycle. Unit: NTUs
ntu	TimeValue	1	attr	Unit measuring all times and providing a constant of the whole network. For level 1, this is always the CAN bit time. Unit: seconds.
operationMode	Boolean	1	attr	Possible operation modes True: Time-Triggered False: Event-Synchronised-Time-Triggered

Table A.938: TtcanCluster

Class	**UdpNmCluster**			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	Udp specific NmCluster attributes			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *NmCluster*, *Referrable*			
Aggregated by	NmConfig.nmCluster			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
nmCbvPosition	Integer	0..1	attr	Defines the position of the control bit vector within the Nm Pdu (Byte position). If this attribute is not configured, the Control Bit Vector is not used.
nmImmediate NmCycleTime	TimeValue	0..1	attr	Defines the immediate NmPdu cycle time in seconds which is used for nmImmediateNmTransmissions NmPdu transmissions. This attribute is only valid if nmImmediate NmTransmissions is greater one.
nmImmediate Nm Transmissions	PositiveInteger	0..1	attr	Defines the number of immediate NmPdus which shall be transmitted. If the value is zero no immediate NmPdus are transmitted. The cycle time of immediate NmPdus is defined by nmImmediateNmCycleTime.
nmMessage TimeoutTime	TimeValue	0..1	attr	Timeout of a NmPdu in seconds. It determines how long the NM shall wait with notification of transmission failure while communication errors occur on the bus.
nmMsgCycle Time	TimeValue	0..1	attr	Period of a NmPdu in seconds. It determines the periodic rate in the periodic transmission mode with bus load reduction and is the basis for transmit scheduling in the periodic transmission mode without bus load reduction.
nmNetwork Timeout	TimeValue	0..1	attr	Network Timeout for NmPdus in seconds. It denotes the time how long the UdpNm shall stay in the Network Mode before transition into Prepare Bus-Sleep Mode shall take place.
nmNidPosition	Integer	0..1	attr	Defines the byte position of the source node identifier within the NmPdu. If this attribute is not configured, the Node Identification is not used.
nmRemote SleepIndication Time	TimeValue	0..1	attr	Timeout for Remote Sleep Indication in seconds. It defines the time how long it shall take to recognize that all other nodes are ready to sleep.
nmRepeat MessageTime	TimeValue	0..1	attr	Timeout for Repeat Message State in seconds. Defines the time how long the NM shall stay in the Repeat Message State.
nmWaitBus SleepTime	TimeValue	0..1	attr	Timeout for bus calm down phase in seconds. It denotes the time how long the CanNm shall stay in the Prepare Bus-Sleep Mode before transition into Bus-Sleep Mode shall take place.
vlan	EthernetPhysical Channel	0..1	ref	Reference to the vlan (represented by the Ethernet PhysicalChannel) this UdpNmCluster shall apply to.

Table A.939: UdpNmCluster

Class	UdpNmNode			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	Udp specific NM Node attributes.			
Base	ARObject, Identifiable, MultilanguageReferrable, NmNode, Referrable			
Aggregated by	NmCluster.nmNode			
Attribute	Type	Mult.	Kind	Note
allNmMessages KeepAwake	Boolean	0..1	attr	Specifies if Nm drops irrelevant NM PDUs.
				false: Only NM PDUs with a Partial Network Information Bit (PNI) = true and containing a Partial Network request for this ECU trigger the standard RX indication handling and thus keep the ECU awake
				true: Every NM PDU triggers the standard RX indication handling and keeps the ECU awake
nmMsgCycle Offset	TimeValue	0..1	attr	Node specific time offset in the periodic transmission node. It determines the start delay of the transmission. Specified in seconds.

Table A.940: UdpNmNode

Class	UdpProps			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	This meta-class specifies the configuration options for UDP (User Datagram Protocol).			
Base	ARObject			
Aggregated by	EthTcpIpProps.udpProps			
Attribute	Type	Mult.	Kind	Note
udpTtl	PositiveInteger	0..1	attr	Default Time-to-live value of outgoing UDP packets.

Table A.941: UdpProps

Class	UdpTp			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Content Model for UDP configuration.			
Base	ARObject, TcpUdpConfig, TransportProtocolConfiguration			
Aggregated by	ApApplicationEndpoint.tpConfiguration, ApplicationEndpoint.tpConfiguration, RtpTp.tcpUdpConfig			
Attribute	Type	Mult.	Kind	Note
udpTpPort	TpPort	1	aggr	Udp Port configuration.

Table A.942: UdpTp

Document ID 635: AUTOSAR_TR_ClassicAutosarModelConstraints

Class	Unit			
Package	M2::MSR::AsamHdo::Units			
Note	This is a physical measurement unit. All units that might be defined should stem from SI units. In order to convert one unit into another factor and offset are defined.			
	For the calculation from SI-unit to the defined unit the factor (factorSiToUnit) and the offset (offsetSiTo Unit) are applied as follows:			
	x [{unit}] := y * [{siUnit}] * factorSiToUnit [[unit]/{siUnit}] + offsetSiToUnit [{unit}]			
	For the calculation from a unit to SI-unit the reciprocal of the factor (factorSiToUnit) and the negation of the offset (offsetSiToUnit) are applied.			
	y {siUnit} := (x*{unit} - offsetSiToUnit [{unit}]) / (factorSiToUnit [[unit]/{siUnit}]			
	Tags:atp.recommendedPackage=Units			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
displayName	SingleLanguageUnit Names	0..1	aggr	This specifies how the unit shall be displayed in documents or in user interfaces of tools.The displayName corresponds to the Unit.Display in an ASAM MCD-2MC file.
				Tags:xml.sequenceOffset=20
factorSiToUnit	Float	0..1	attr	This is the factor for the conversion from SI Units to units.
				The inverse is used for conversion from units to SI Units.
				Tags:xml.sequenceOffset=30
offsetSiToUnit	Float	0..1	attr	This is the offset for the conversion from and to siUnits.
				Tags:xml.sequenceOffset=40
physical Dimension	PhysicalDimension	0..1	ref	This association represents the physical dimension to which the unit belongs to. Note that only values with units of the same physical dimensions might be converted.
				Tags:xml.sequenceOffset=50

Table A.943: Unit

Primitive	UnlimitedInteger
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::PrimitiveTypes
Note	An instance of UnlimitedInteger is an element in the set of integer numbers (..., -2, -1, 0, 1, 2, ...).
	The range is limited by constraint 2534.
	The value can be expressed in decimal, octal, hexadecimal and binary representation. Negative numbers can only be expressed in decimal notation.
	Tags: xml.xsd.customType=UNLIMITED-INTEGER xml.xsd.pattern=0|[\+\-]?[1-9][0-9]*|0[xX][0-9a-fA-F]+|0[bB][0-1]+|0[0-7]+ xml.xsd.type=string

Table A.944: UnlimitedInteger

Class	UserDefinedIPdu			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	UserDefinedIPdu allows to describe PDU-based communication over Complex Drivers. If a new BSW module is added above the PduR (e.g. a Diagnostic Service) then this IPdu element shall be used to describe the communication. **Tags:**atp.recommendedPackage=Pdus			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *IPdu*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Pdu*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
cddType	String	0..1	attr	This attribute defines the CDD that transmits or receives the UserDefinedPdu. If several CDDs are defined this attribute is used to distinguish between them.

Table A.945: UserDefinedIPdu

Class	UserDefinedPdu			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	UserDefinedPdu allows to describe PDU-based communication over Complex Drivers. If a new BSW module is added above the BusIf (e.g. a new Nm module) then this Pdu element shall be used to describe the communication. **Tags:**atp.recommendedPackage=Pdus			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Pdu*, *Referrable*			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
cddType	String	0..1	attr	This attribute defines the CDD that transmits or receives the UserDefinedIPdu. If several CDDs are defined this attribute is used to distinguish between them.

Table A.946: UserDefinedPdu

Class	UserDefinedPhysicalChannel			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::CddSupport			
Note	This element allows the modeling of arbitrary Physical Channels.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *PhysicalChannel*, *Referrable*			
Aggregated by	*CommunicationCluster*.physicalChannel			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

Table A.947: UserDefinedPhysicalChannel

Class	*ValueSpecification* (abstract)
Package	M2::AUTOSARTemplates::CommonStructure::Constants
Note	Base class for expressions leading to a value which can be used to initialize a data object.
Base	*ARObject*
Subclasses	*AbstractRuleBasedValueSpecification*, ApplicationValueSpecification, *CompositeValueSpecification*, ConstantReference, NotAvailableValueSpecification, NumericalValueSpecification, ReferenceValue Specification, TextValueSpecification

▽

△

Class	**ValueSpecification** (abstract)			
Aggregated by	ApplicationAssocMapElementValueSpecification.key, ApplicationAssocMapElementValueSpecification. value, ArrayValueSpecification.element, CalibrationParameterValue.applInitValue, CalibrationParameter Value.implInitValue, ConstantSpecification.valueSpec, CryptoServiceKey.developmentValue, Diagnostic EnvDataCondition.compareValue, DiagnosticEnvDataElementCondition.compareValue, FieldSenderCom Spec.initValue, ISignal.initValue, ISignal.timeoutSubstitutionValue, NonqueuedReceiverComSpec.init Value, NonqueuedReceiverComSpec.timeoutSubstitutionValue, NonqueuedSenderComSpec.initValue, NvProvideComSpec.ramBlockInitValue, NvProvideComSpec.romBlockInitValue, NvRequireComSpec.init Value, ParameterDataPrototype.initValue, ParameterProvideComSpec.initValue, ParameterRequireCom Spec.initValue, PersistencyDataRequiredComSpec.initValue, PersistencyKeyValuePair.initValue, Port DefinedArgumentValue.value, PortPrototypeBlueprintInitValue.value, RecordValueSpecification.field, *StateManagementCompareCondition*.compareValue, SwDataDefProps.invalidValue, VariableData Prototype.initValue			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
shortLabel	Identifier	0..1	attr	This can be used to identify particular value specifications for human readers, for example elements of a record type.

Table A.948: ValueSpecification

Class	**VariableAccess**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::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*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	*AtpClassifier*.atpFeature, *ReceiverComSpec*.replaceWith, RunnableEntity.dataReadAccess, Runnable Entity.dataReceivePointByArgument, RunnableEntity.dataReceivePointByValue, RunnableEntity.data SendPoint, RunnableEntity.dataWriteAccess, RunnableEntity.readLocalVariable, RunnableEntity.written LocalVariable			
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 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 A.949: VariableAccess

Class	**VariableAndParameterInterfaceMapping**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	Defines the mapping of VariableDataPrototypes or ParameterDataPrototypes in context of two different SenderReceiverInterfaces, NvDataInterfaces or ParameterInterfaces.			
Base	*ARObject*, *AtpBlueprint*, *AtpBlueprintable*, *Identifiable*, *MultilanguageReferrable*, *PortInterfaceMapping*, *Referrable*			
Aggregated by	PortInterfaceMappingSet.portInterfaceMapping			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
dataMapping	DataPrototypeMapping	*	aggr	Defines the mapping of two particular VariableData Prototypes or ParameterDataPrototypes with unequal names and/or unequal semantic (resolution or range) in context of two different SenderReceiverInterfaces, Nv DataInterfaces or ParameterInterfaces

Table A.950: VariableAndParameterInterfaceMapping

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.
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *AutosarDataPrototype*, *DataPrototype*, *Identifiable*, *Multilanguage Referrable*, *Referrable*
Aggregated by	ApplicationInterface.indication, *AtpClassifier*.atpFeature, BswInternalBehavior.arTypedPerInstance Memory, BswModuleDescription.providedData, BswModuleDescription.requiredData, BulkNvData Descriptor.bulkNvBlock, *InternalBehavior*.staticMemory, NvBlockDescriptor.ramBlock, NvDataInterface. nvData, SenderReceiverInterface.dataElement, ServiceInterface.event, SwcInternalBehavior.arTypedPer InstanceMemory, SwcInternalBehavior.explicitInterRunnableVariable, SwcInternalBehavior.implicitInter RunnableVariable

Attribute	Type	Mult.	Kind	Note
initValue	ValueSpecification	0..1	aggr	Specifies initial value(s) of the VariableDataPrototype

Table A.951: VariableDataPrototype

Class	VariableDataPrototypeInSystemInstanceRef
Package	M2::AUTOSARTemplates::SystemTemplate::InstanceRefs
Note	
Base	*ARObject*, *AtpInstanceRef*
Aggregated by	EndToEndProtectionVariablePrototype.receiver, EndToEndProtectionVariablePrototype.sender, Port ElementToCommunicationResourceMapping.variableDataPrototype, SenderReceiverCompositeElement ToSignalMapping.dataElement, SenderReceiverToSignalGroupMapping.dataElement, SenderReceiverTo SignalMapping.dataElement, SignalServiceTranslationEventProps.translationTarget, SwcToSwcSignal. dataElement

Attribute	Type	Mult.	Kind	Note
base	System	0..1	ref	**Stereotypes:** atpDerived
context Component (ordered)	SwComponent Prototype	*	ref	
context Composition	RootSwComposition Prototype	0..1	ref	
contextPort	PortPrototype	1	ref	
targetData Prototype	VariableDataPrototype	0..1	ref	

Table A.952: VariableDataPrototypeInSystemInstanceRef

Class	VariableInAtomicSWCTypeInstanceRef
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::DataElements::InstanceRefs Usage
Note	
Base	*ARObject*, *AtpInstanceRef*
Aggregated by	AutosarVariableRef.autosarVariable

Attribute	Type	Mult.	Kind	Note
base	AtomicSwComponent Type	0..1	ref	**Stereotypes:** atpDerived **Tags:**xml.sequenceOffset=10
contextData Prototype (ordered)	ApplicationComposite ElementDataPrototype	*	ref	This is the context in a compositeDataType. **Tags:**xml.sequenceOffset=40

▽

△

Class	VariableInAtomicSWCTypeInstanceRef			
portPrototype	PortPrototype	0..1	ref	This is the port providing the parameter or the entry point to the parameter structure. **Tags:**xml.sequenceOffset=20
rootVariable DataPrototype	VariableDataPrototype	0..1	ref	**Tags:**xml.sequenceOffset=30
targetData Prototype	DataPrototype	0..1	ref	This is the target of the instance ref. Note that it shall be one of ApplicationCompositeElementDataPrototype of VariableDataPrototype. **Tags:**xml.sequenceOffset=50

Table A.953: VariableInAtomicSWCTypeInstanceRef

Class	VariationPoint			
Package	M2::AUTOSARTemplates::GenericStructure::VariantHandling			
Note	This meta-class represents the ability to express a "structural variation point". The container of the variation point is part of the selected variant if swSyscond evaluates to true and each postBuildVariant Criterion is fulfilled.			
Base	ARObject			
Attribute	Type	Mult.	Kind	Note
blueprint Condition	DocumentationBlock	0..1	aggr	This represents a description that documents how the variation point shall be resolved when deriving objects from the blueprint. Note that variationPoints are not allowed within a blueprintCondition. **Tags:**xml.sequenceOffset=28
desc	MultiLanguageOverview Paragraph	0..1	aggr	This allows to describe shortly the purpose of the variation point. **Tags:**xml.sequenceOffset=20
formalBlueprint Generator	BlueprintGenerator	0..1	aggr	This represents a description that documents how the variation point shall be resolved when deriving objects from the blueprint by using ARMQL. Note that variationPoints are not allowed within a formal BlueprintGenerator. **Tags:** atp.Status=draft xml.sequenceOffset=30
postBuildVariant Condition	PostBuildVariant Condition	*	aggr	This is the set of post build variant conditions which all shall be fulfilled in order to (postbuild) bind the variation point. **Tags:**xml.sequenceOffset=40
sdg	Sdg	0..1	aggr	An optional special data group is attached to every variation point. These data can be used by external software systems to attach application specific data. For example, a variant management system might add an identifier, an URL or a specific classifier. **Tags:**xml.sequenceOffset=50

▽

△

Class	VariationPoint			
shortLabel	Identifier	0..1	attr	This provides a name to the particular variation point to support the RTE generator. It is necessary for supporting splitable aggregations and if binding time is later than codeGenerationTime, as well as some RTE conditions. It needs to be unique with in the enclosing Identifiables with the same ShortName. **Stereotypes:** atpIdentityContributor **Tags:**xml.sequenceOffset=10
swSyscond	ConditionByFormula	0..1	aggr	This condition acts as Binding Function for the Variation Point. Note that the multiplicity is 0..1 in order to support pure postBuild variants. **Tags:**xml.sequenceOffset=30

Table A.954: VariationPoint

Class	VariationPointProxy			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::VariantHandling			
Note	The VariationPointProxy represents variation points of the C/C++ implementation. In case of bindingTime = compileTime the RTE provides defines which can be used for Pre Processor directives to implement compileTime variability.			
Base	ARObject, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	BswInternalBehavior.variationPointProxy, SwcInternalBehavior.variationPointProxy			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
conditionAccess	ConditionByFormula	0..1	aggr	This condition acts as Binding Function for the Variation Point.
implementation DataType	AbstractImplementation DataType	0..1	ref	This association to ImplementationDataType shall be taken as an implementation hint by the RTE generator.
postBuildValue Access	PostBuildVariant Criterion	0..1	ref	This represents the applicable PostBuildVariantCriterion in the context of a VariationPointProxy. Note that the technical details how to access the particular postBuildValueAccess are still considered internal to the RTE and are consequently not standardized.
postBuildVariant Condition	PostBuildVariant Condition	*	aggr	This represents that applicable PostBuoldVariant Condition in the context of aVariationPointProxy.
valueAccess	AttributeValueVariation Point	0..1	aggr	This value acts as Binding Function for the VariationPoint.

Table A.955: VariationPointProxy

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:**atp.recommendedPackage=TimingExtensions			
Base	*ARElement*, ARObject, *AtpBlueprint*, AtpBlueprintable, CollectableElement, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*, *TimingExtension*			
Aggregated by	ARPackage.element			
Attribute	**Type**	**Mult.**	**Kind**	**Note**

▽

△

Class	VfbTiming			
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 A.956: VfbTiming

Class	VlanConfig			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	VLAN Configuration attributes			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Aggregated by	EthernetPhysicalChannel.vlan			
Attribute	Type	Mult.	Kind	Note
vlanIdentifier	PositiveInteger	1	attr	A VLAN is identified by this attribute according to IEEE 802.1Q. The allowed values range is from 0..4095.

Table A.957: VlanConfig

Class	VlanMembership			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Static logical channel or VLAN binding to a switch-port.			
	The reference to an EthernetPhysicalChannel without a VLAN defined represents the handling of untagged frames.			
Base	ARObject			
Aggregated by	CouplingPort.vlanMembership			
Attribute	Type	Mult.	Kind	Note
defaultPriority	PositiveInteger	1	attr	Standard output-priority outgoing Frames will be tagged with.
				Defines the priority that received frames are assigned together with the VLAN Id (defaultVlan). The values from 0 (best effort) to 7 (highest) are allowed.
				In case modifyVlan and an already tagged received frame, the actual priority of the received frame is not modified.
dhcpAddress Assignment	DhcpServer Configuration	0..1	aggr	Specifies the IP Address which will be assigned to a DHCP Client at this SwitchPort. If no dhcpAddress Assignment is provided all DHCP-Discover messages received at this Port will be discarded by the DHCP Server.
sendActivity	EthernetSwitchVlan EgressTaggingEnum	0..1	attr	Attribute denotes whether a VLAN tagged ethernet frame will be
				1. sent with its VLAN tag (sentTagged)
				2. sent without a VLAN tag (sentUntagged)
				3. will be dropped at this port (notSent or VLAN not member of this list)
vlan	EthernetPhysical Channel	1	ref	References a channel that represents a VLAN or an untagged channel.

Table A.958: VlanMembership

Class	WaitPoint			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents			
Note	This defines a wait-point for which the RunnableEntity can wait.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	RunnableEntity.waitPoint			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
timeout	TimeValue	0..1	attr	Time in seconds before the WaitPoint times out and the blocking wait call returns with an error indicating the timeout.
trigger	RTEEvent	0..1	ref	This is the RTEEvent this WaitPoint is waiting for.

Table A.959: WaitPoint

Class	WorstCaseHeapUsage			
Package	M2::AUTOSARTemplates::CommonStructure::ResourceConsumption::HeapUsage			
Note	Provides a formal worst case heap usage.			
Base	*ARObject*, *HeapUsage*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Aggregated by	ResourceConsumption.heapUsage			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
memory Consumption	PositiveInteger	0..1	attr	Worst case heap consumption. Unit: byte.

Table A.960: WorstCaseHeapUsage

Class	WorstCaseStackUsage			
Package	M2::AUTOSARTemplates::CommonStructure::ResourceConsumption::StackUsage			
Note	Provides a formal worst case stack usage.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *StackUsage*			
Aggregated by	ResourceConsumption.stackUsage			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
memory Consumption	PositiveInteger	0..1	attr	Worst case stack consumption. Unit: byte.

Table A.961: WorstCaseStackUsage

Class	Xdoc			
Package	M2::MSR::Documentation::TextModel::InlineTextElements			
Note	This meta-class represents the ability to refer to an external document which can be rendered as printed matter.			
Base	*ARObject*, *Referrable*, *SingleLanguageReferrable*			
Aggregated by	MixedContentForParagraph.xdoc			
Attribute	**Type**	**Mult.**	**Kind**	**Note**
date	DateTime	0..1	attr	This element specifies the release date of the external document if applicable. **Tags:**xml.sequenceOffset=50
number	String	0..1	attr	This represents document number of an external document that is referenced. Kept as a string. **Tags:**xml.sequenceOffset=30

$\bigtriangledown$

Document ID 635: AUTOSAR_TR_ClassicAutosarModelConstraints

△

Class	Xdoc			
position	String	0..1	attr	This represents the reference to the relevant positions of a standard. Kept as a string. **Tags:**xml.sequenceOffset=80
publisher	String	0..1	attr	This represents the publisher of an external document that is being referenced. Kept as a string. **Tags:**xml.sequenceOffset=60
state	String	0..1	attr	This represents version and state of the external document. Kept as a string. **Tags:**xml.sequenceOffset=40
url	Url	0..1	aggr	This specifies the URL of the external document. **Tags:**xml.sequenceOffset=70

Table A.962: Xdoc

Class	Xfile			
Package	M2::MSR::Documentation::TextModel::InlineTextElements			
Note	This represents to reference an external file within a documentation.			
Base	*ARObject*, *Referrable*, *SingleLanguageReferrable*			
Aggregated by	*MixedContentForParagraph*.xfile			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
tool	String	0..1	attr	This element describes the tool which was used to generate the corresponding Xfile . Kept as a string since no specific syntax can be provided to denote a tool. **Tags:**xml.sequenceOffset=50
toolVersion	String	0..1	attr	This element describes the tool version which was used to generate the corresponding xfile. Kept as a string, since no specific syntax can be specified. **Tags:**xml.sequenceOffset=60
url	Url	0..1	aggr	This represents the URL of the external file. **Tags:**xml.sequenceOffset=30

Table A.963: Xfile

Class	XrefTarget			
Package	M2::MSR::Documentation::TextModel::InlineTextElements			
Note	This element specifies a reference target which can be scattered throughout the text.			
Base	*ARObject*, *Referrable*, *SingleLanguageReferrable*			
Aggregated by	*MixedContentForOverviewParagraph*.xrefTarget, *MixedContentForParagraph*.xrefTarget			
Attribute	*Type*	*Mult.*	*Kind*	*Note*
–	–	–	–	–

Table A.964: XrefTarget