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

Document Status	Final
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	4.4.0

Document Change History			
Date	**Release**	**Changed by**	**Description**
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

Table of Contents

References

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

[2] List of Basic Software Modules
AUTOSAR_TR_BSWModuleList

[3] Software Component Template
AUTOSAR_TPS_SoftwareComponentTemplate

[4] Specification of RTE Software
AUTOSAR_SWS_RTE

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

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

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

[8] ISO 17356-3: Road vehicles – Open interface for embedded automotive applications – Part 3: OSEK/VDX Operating System (OS)

[9] Collection of blueprints for AUTOSAR M1 models
AUTOSAR_MOD_GeneralBlueprints

[10] Generic Structure Template
AUTOSAR_TPS_GenericStructureTemplate

[11] Specifications of Safety Extensions
AUTOSAR_TPS_SafetyExtensions

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

[13] Specification of COM Based Transformer
AUTOSAR_SWS_COMBasedTransformer

[14] SAE J1939-21 Data Link Layer

1 Document Information and Content

This auxiliary document provides a collection of constraints for AUTOSAR models. All constraints are copied from template specification and software specification documents, 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.

The referenced deliverable AUTOSAR_SWS_LINNetworkManagement is set to status "obsolete" in release 4.4.0.

2 Autosar Model Constraints

2.1 ASWS_TransformerGeneral

[SWS_Xfrm_CONSTR_09094] ⌈ If there exists a `XfrmImplementationMapping` which references an `ISignal` or `ISignalGroup` *sig1* and contains the optional parameter `XfrmVariableDataPrototypeInstanceRef` , all `XfrmImplementationMapping` s which reference the same `ISignal` or `ISignalGroup` *sig1* shall contain a `XfrmVariableDataPrototypeInstanceRef` . ⌋*(SRS_Xfrm_00001)*

[SWS_Xfrm_CONSTR_09095] ⌈ The `XfrmVariableDataPrototypeInstanceRef` shall refer to the instance of a `VariableDataPrototype` which belongs to a subclass of an `AtomicSwComponentType` . ⌋*(SRS_Xfrm_00001)*

[SWS_Xfrm_CONSTR_09096] ⌈ If no `XfrmSignal` exists and hence no `ISignal` or `ISignalGroup` is referenced, `XfrmVariableDataPrototypeInstanceRef` shall be used to reference the instance of the `VariableDataPrototype` which data shall be transformed. ⌋*(SRS_Xfrm_00001)*

2.2 SWS_ADCDriver

[constr_SWS_Adc_CONSTR_00001] DRAFT ⌈ The ECUC partitions referenced by AdcKernelEcucPartitionRef shall be a subset of the ECUC partitions referenced by Adc EcucPartitionRef.

⌋*()*

[constr_SWS_Adc_CONSTR_00002] DRAFT ⌈ The ECUC partitions referenced by AdcGroupEcucPartitionRef shall be a subset of the ECUC partitions referenced by Adc EcucPartitionRef.

⌋*()*

2.3 SWS_BSWModeManager

[constr_SWS_BswM_CONSTR_00001] ⌈ The BswM shall reject configurations where a BswMActionList contains BswMActionListItems with same-valued BswMActionList ItemIndexes.

⌋()

[constr_SWS_BswM_CONSTR_00002] ⌈ The value of CompuMethod.category referenced by the foreign reference of BswMCompuMethodRef shall be TEXTTABLE.

⌋()

[constr_SWS_BswM_CONSTR_00003] ⌈ The BswM shall reject configurations where a BswMDeadlineMonitoringControl container has a BswMDisabledDMPduGroupRef and a BswMEnabledDMPduGroupRef which reference the same PDU Group.

⌋()

[constr_SWS_BswM_CONSTR_00004] ⌈ The BswM shall reject configurations where a BswMPduGroupSwitch container has a BswMDisabledPduGroupRef and a Bsw MEnabledPduGroupRef which reference the same PDU Group.

⌋()

2.4 SWS_BusMirroring

[SWS_Mirror_CONSTR_00001] ⌈ The `MirrorDestPdu` of a `MirrorDestNetworkCan` requires a `MetaDataItem` of `MetaDataItemType` `CAN_ID_32` . The `CanIfTxPduCanIdMask` of the corresponding `CanIfTxPduCfg` shall be `0` . ⌋ *(SRS_Mirror_00001)*

[SWS_Mirror_CONSTR_00002] ⌈ The `CanFdPaddingValue` that is used to transmit the PDU referenced by `MirrorDestPduRef` for a CAN-FD destination bus shall be set to `0` to ensure that the `NetworkStateAvailable` of a CAN status item is `0` if the status item has not been written by the

but lies in a padded region of the status frame. ⌋*(SRS_Mirror_00001)*

[SWS_Mirror_CONSTR_00003] ⌈ The configured `MirrorSourceFlexRayFilter` s shall be configured such that they do not include serialized frames transmitted on the source bus. ⌋*(SRS_Mirror_00001)*

[SWS_Mirror_CONSTR_00004] ⌈ `FrIfAllowDynamicLSduLength` shall be set to true for all `FrIfFrameStructure` s that contain `FrIfTxPdu` s referenced by a `MirrorDestPdu` of a `MirrorDestNetworkFlexRay` . ⌋*(SRS_Mirror_00001)*

2.5 SWS_CANDriver

[constr_SWS_Can_CONSTR_00508] DRAFT ⌈ The module will operate as an independent instance in each of the partitions, means the called API will only target the partition it is called in.

⌋()

[constr_SWS_Can_CONSTR_00509] DRAFT ⌈ The ECUC partitions referenced by CanControllerEcucPartitionRef shall be a subset of the ECUC partitions referenced by CanEcucPartitionRef.

⌋()

[constr_SWS_Can_CONSTR_00510] DRAFT ⌈ CanController and CanTrcvChannel of one communication channel shall all reference the same ECUC partition.

⌋()

2.6 SWS_COMManager

[constr_SWS_ComM_CONSTR_00001] ⌈ ComM channel's that are referenced by a PNC are not allowed to be referenced by any ComMUsers, if the PNC references at least one EthIfSwitchPortGroup (see figure [REF] Use Case 6). A configuration tool shall reject such a configuration as invalid (error). This constraint is only valid for a host ecu that control an Ethernet switch. In all other UseCases ComMChannels can be referenced by a PNC's and ComMUsers.

⌋()

2.7 SWS_CryptoDriver

[constr_SWS_Crypto_CONSTR_00001] Draft ⌈ The Crypto Driver module will operate as an independent instance in each of the partitions, means the called API will only target the partition it is called in.

⌋()

[constr_SWS_Crypto_CONSTR_00002] Draft ⌈ The ECUC partitions referenced by CryptoDriverObjectEcucPartitionRef shall be a subset of the ECUC partitions referenced by CryptoEcucPartitionRef.

⌋()

[constr_SWS_Crypto_CONSTR_00003] Draft ⌈ If the CryptoDriverObjectEcucPartitionRef shall be configured for an HSM it shall be mapped to 0 or 1 ECUC partitions only.

⌋()

2.8 SWS_DiagnosticCommunicationManager

[SWS_Dcm_CONSTR_6000] Harmonize the naming between interfaces and modes ⌈ The shortname of `DcmDspSessionRow` shall match names of Dcm_SesCtrlType and of the mode declarations of DcmDiagnosticSessionControl. The "DCM_" prefix is mandatory for all shortnames. ⌋()

[SWS_Dcm_CONSTR_6001] Provide standardized names for ISO standardized diagnostic sessions ⌈ The following values of `DcmDspSessionLevel` which represent ISO defined diagnostic sessions shall be used for the shortname of `DcmDspSessionRow` :
1 DCM_DEFAULT_SESSION
2 DCM_PROGRAMMING_SESSION
3 DCM_EXTENDED_DIAGNOSTIC_SESSION
4 DCM_SAFETY_SYSTEM_DIAGNOSTIC_SESSION
⌋()

[SWS_Dcm_CONSTR_6002] Existence of size parameter ⌈ `DcmDspDataByteSize` shall be present if `DcmDspDataType` is set to: UINT8_N, SINT8_N, UINT16_N, SINT16_N, UINT32_N, SINT32_N or UINT8_DYN. ⌋()

[SWS_Dcm_CONSTR_6008] Define the usage of `DcmDspRoutineParameterSize` parameter ⌈ `DcmDspRoutineParameterSize` is only required if `DcmDspRoutineSignalType` is set to SINT8_N, SINT16_N, SINT32_N, UINT8_N, UINT16_N, UINT32_N or VARIABLE_LENGTH. ⌋()

[SWS_Dcm_CONSTR_6011] Only last parameters in `RID` may have a variable length ⌈ DcmDspRoutineSignalType with VARIABLE_LENGTH is only valid for the last signal. ⌋()

[SWS_Dcm_CONSTR_6012] Existence of size parameter ⌈ `DcmDspPidDataByteSize` shall be present if `DcmDspPidDataType` is set to: UINT8_N, SINT8_N, UINT16_N, SINT16_N, UINT32_N or SINT32_N. ⌋()

[SWS_Dcm_CONSTR_6018] ⌈ `DcmDspData` elements used in service 0x2E shall not have DcmDspDataUsePorts set to USE_ECU_SIGNAL. ⌋()

[SWS_Dcm_CONSTR_6020] Definition of allowed `DID` access ⌈ Any defined range shall only reference via `DcmDspDidRangeInfoRef` . The sub-containers `DcmDspDidControl` and DcmDspDidDefineinDcmDspDidInfo shall not be used] . ⌋()

[SWS_Dcm_CONSTR_6021] DID ranges cannot be mapped on DDDIDs, because service 0x2C DDDID does not support the range feature. Practically `DcmDspDidRangeIdentifierLowerLimit` and `DcmDspDidRangeIdentifierUpperLimit` should not include DIDs of the range 0xF200 till 0xF3FF. ⌈ Any

defined range shall only reference DcmDspDidInfo via DcmDspDidRangeInfoRef , having set DcmDspDidDynamicallyDefined == False. ⌋()

[SWS_Dcm_CONSTR_6023] DcmDspDidRef shall not reference the same DID reference twice ⌈ DcmDspDid container shall not include the same DcmDspDidRef parameters more than once. ⌋()

[SWS_Dcm_CONSTR_6025] Reference to DcmDslResponseOnEvent connection ⌈ Only one DcmDslROEConnectionRef shall reference DcmDslResponseOnEvent connection. ⌋()

[SWS_Dcm_CONSTR_6026] Usage of variable data length in case of S/R communication, NvRam access or ECU signal access ⌈ In case DcmDspDataUsePort is set to { USE_DATA_SENDER_RECEIVER , USE_DATA_SENDER_RECEIVER_AS_SERVICE , USE_BLOCK_ID , USE_ECU_SIGNAL }, the usage of variable data length shall be not allowed. ⌋ ()

[SWS_Dcm_CONSTR_6027] ⌈ The application will inform the Dcm by calling Xxx_Set ActiveDiagnostic() about the ActiveDiagnostic status. ⌋()

[SWS_Dcm_CONSTR_6028] ⌈ DcmModeCondition shall either have a DcmBswModeRef or a DcmSwcModeRef or a DcmSwcSRDataElementRef as external reference. ⌋()

[SWS_Dcm_CONSTR_6029] ⌈ The values DCM_GREATER_THAN, DCM_GREATER_OR_EQUAL, DCM_LESS_OR_EQUAL and DCM_LESS_THAN shall not used with a Mode reference (DcmBswModeRef or DcmSwcModeRef) . ⌋()

[SWS_Dcm_CONSTR_6030] ⌈ The ReturnControlToEcu functionnality is existing if at least one of the following parameters are activated : DcmDspDidFreezeCurrentState in ECUC_Dcm_00624 : or DcmDspDidResetToDefault in ECUC_Dcm_00623 : or DcmDspDidShortTermAdjustment in ECUC_Dcm_00625 : . ⌋()

[SWS_Dcm_CONSTR_6031] ⌈ The DcmDspData .SHORT-NAME and DcmDspPidData .SHORT-NAME shall be distinct. ⌋()

[SWS_Dcm_CONSTR_6035] Restrictions on size parameter for 16 Bit arrays ⌈ DcmDspDataByteSize shall be a multiple of 2 if the value is greater than 2 and DcmDspDataType is UINT16_N or SINT16_N. ⌋()

[SWS_Dcm_CONSTR_6036] Restrictions on size parameter for 32 Bit arrays ⌈ DcmDspDataByteSize shall be a multiple of 4 if the value is greater than 4 and DcmDspDataType is UINT32_N or SINT32_N. ⌋()

[SWS_Dcm_CONSTR_6038] Restrictions on datatype usage ⌈ DcmDspDataType shall be UINT8_N, in case DcmDspDataUsePort is equal to USE_BLOCK_ID . ⌋()

[SWS_Dcm_CONSTR_6039] Signals with variable datalength ⌈ Only the last signal (DcmDspDidSignal) of a DID can have variable datalength (DcmDspDataType is set to UINT8_DYN). ⌋()

[SWS_Dcm_CONSTR_6040] Restrictions on size parameter for 16 Bit arrays ⌈ DcmDspPidDataByteSize shall be a multiple of 2 if the value is greater than 2 and DcmDspPIDDataType is UINT16_N or SINT16_N. ⌋*()*

[SWS_Dcm_CONSTR_6041] Restrictions on size parameter for 32 Bit arrays ⌈ DcmDspPidDataByteSize shall be a multiple of 4 if the value is greater than 4 and DcmDspPIDDataType is UINT32_N or SINT32_N. ⌋*()*

[SWS_Dcm_CONSTR_6044] ⌈ Generic connections shall be consistent. This means that the MetaDataItems and the PduLength of all referenced PDUs of a DcmDslConnection (DcmDslProtocolRxPduRef , DcmDslProtocolTxPduRef , DcmDslPeriodicTxPduRef , DcmDslRoeTxPduRef)are identical. ⌋*()*

[SWS_Dcm_CONSTR_6045] ⌈ In case the responsibility is on provider side (DcmDspVehInfoNODIProvResp is set to TRUE), only one DcmDspVehInfoData container shall be allowed. ⌋*()*

[SWS_Dcm_CONSTR_6046] ⌈ In case DcmDspVehInfoDataUsePort is set to FALSE and DcmDspVehInfoDataReadFnc is set to either Dem_DcmGetInfoType Value08 or Dem_DcmGetInfoTypeValue0B then DcmDspVehInfoNODIProvResp shall be set to TRUE. ⌋*()*

[SWS_Dcm_CONSTR_6047] ⌈ Id of the Service identifier configured in DcmDsdSidTabServiceId shall be unique within one DcmDsdServiceTable . ⌋*()*

[SWS_Dcm_CONSTR_6048] Composite sub elements accessible only by read ⌈ Composite sub elements can only be referred from Read DID i.e. Write and Control DID are not supported. ⌋*()*

[SWS_Dcm_CONSTR_6050] ⌈ If a DcmDspDid is used in service 0x2F and is configured to have an atomic S/R interface, the DcmDspDidControlMask shall be set to DCM_CONTROLMASK_EXTERNAL and the parameter DcmDspDidControlMaskSize shall be present with a value greater than zero. ⌋*()*

[SWS_Dcm_CONSTR_6051] ⌈ The configuration parameter DcmDspDidControlMaskSize shall be only present if DcmDspDidControlMask is equal to DCM_CONTROLMASK_EXTERNAL or DCM_CONTROLMASK_INTERNAL . ⌋*()*

[SWS_Dcm_CONSTR_6053] ⌈ The aggregation of DcmDspTextTableMapping at DcmDspAlternativeDataType is only valid if the category of the CompuMethod of the DataType referenced by DcmDspAlternativeDataType.DcmApplicationDataType has category set to TEXTTABLE or SCALE_LINEAR_AND_TEXTTABLE. ⌋*()*

[SWS_Dcm_CONSTR_6054] Existence of DTCStatusMask ⌈ DcmDspRoeDTCStatusMask shall be present if DcmDspRoeInitialEventStatus is set to DCM_ROE_STOPPED . ⌋*()*

[SWS_Dcm_CONSTR_6055] Dependency for DcmDslProtocolMaximumResponseSize ⌈ DcmDslProtocolMaximumResponseSize shall be only present if DcmPagedBufferEnabled is set to TRUE. ⌋*()*

[SWS_Dcm_CONSTR_6056] Dependency for `DcmDslProtocolTransType` ⌈ `DcmDslProtocolTransType` shall be only present if the `Dcm_ProtocolType` is configured to `DCM_ROE_ON_CAN` or `DCM_ROE_ON_FLEXRAY` or `DCM_ROE_ON_IP` . ⌋ *()*

[SWS_Dcm_CONSTR_6057] Dependency for `DcmDspDataEcuSignal` ⌈ `DcmDspDataEcuSignal` shall be only present if `DcmDspDataUsePort` is set to `USE_ECU_SIGNAL` . ⌋*()*

[SWS_Dcm_CONSTR_6058] Dependency for `DcmDspDataEndianness` ⌈ In case `DcmDspDataEndianness` is not configured, the `DcmDspDataDefaultEndianness` shall be used instead. ⌋*()*

[SWS_Dcm_CONSTR_6059] Dependency for `DcmDspDataFreezeCurrentStateFnc` ⌈ `DcmDspDataFreezeCurrentStateFnc` shall be only present if:

- `DcmDspDataUsePort` is set to `USE_DATA_SYNCH_FNC` or
- `DcmDspDataUsePort` is set to `USE_DATA_ASYNCH_FNC` or
- `DcmDspDataUsePort` is set to `USE_DATA_ASYNCH_FNC_ERROR`

⌋*()*

[SWS_Dcm_CONSTR_6060] Dependency for `DcmDspDataGetScalingInfoFnc` ⌈ `DcmDspDataGetScalingInfoFnc` shall be only present if:

- `DcmDspDataUsePort` is set to `USE_DATA_SYNCH_FNC` or
- `DcmDspDataUsePort` is set to `USE_DATA_ASYNCH_FNC` or
- `DcmDspDataUsePort` is set to `USE_DATA_ASYNCH_FNC_ERROR`

⌋*()*

[SWS_Dcm_CONSTR_6061] Dependency for `DcmDspDataReadDataLengthFnc` ⌈ `DcmDspDataReadDataLengthFnc` shall be only present if:

- `DcmDspDataUsePort` is set to `USE_DATA_SYNCH_FNC` or
- `DcmDspDataUsePort` is set to `USE_DATA_ASYNCH_FNC` or
- `DcmDspDataUsePort` is set to `USE_DATA_ASYNCH_FNC_ERROR`

⌋*()*

[SWS_Dcm_CONSTR_6062] Dependency for `DcmDspDataReadFnc` ⌈ `DcmDspDataReadFnc` shall be only present if:

- `DcmDspDataUsePort` is set to `USE_DATA_SYNCH_FNC` or
- `DcmDspDataUsePort` is set to `USE_DATA_ASYNCH_FNC` or
- `DcmDspDataUsePort` is set to `USE_DATA_ASYNCH_FNC_ERROR`

⌋()

[SWS_Dcm_CONSTR_6063] Dependency for `DcmDspDataResetToDefaultFnc` ⌈ `DcmDspDataResetToDefaultFnc` shall be only present if:

- `DcmDspDataUsePort` is set to `USE_DATA_SYNCH_FNC` or
- `DcmDspDataUsePort` is set to `USE_DATA_ASYNCH_FNC` or
- `DcmDspDataUsePort` is set to `USE_DATA_ASYNCH_FNC_ERROR`

⌋()

[SWS_Dcm_CONSTR_6064] Dependency for `DcmDspDidControlMaskSize` ⌈ `DcmDspDidControlMaskSize` shall be only present if `DcmDspDidControlMask` is equal to `DCM_CONTROLMASK_EXTERNAL` or `DCM_CONTROLMASK_INTERNAL` . ⌋()

[SWS_Dcm_CONSTR_6065] Dependency for `DcmDspDataReturnControlToEcuFnc` ⌈ `DcmDspDataReturnControlToEcuFnc` shall be only present if:

- `DcmDspDataUsePort` is set to `USE_DATA_SYNCH_FNC` or
- `DcmDspDataUsePort` is set to `USE_DATA_ASYNCH_FNC` or
- `DcmDspDataUsePort` is set to `USE_DATA_ASYNCH_FNC_ERROR`

⌋()

[SWS_Dcm_CONSTR_6066] Dependency for `DcmDspDataShortTermAdjustmentFnc` ⌈ `DcmDspDataShortTermAdjustmentFnc` shall be only present if:

- `DcmDspDataUsePort` is set to `USE_DATA_SYNCH_FNC` or
- `DcmDspDataUsePort` is set to `USE_DATA_ASYNCH_FNC` or
- `DcmDspDataUsePort` is set to `USE_DATA_ASYNCH_FNC_ERROR`

⌋()

[SWS_Dcm_CONSTR_6067] Dependency for `DcmDspDataBlockIdRef` ⌈ `DcmDspDataBlockIdRef` shall be only present if `DcmDspDataUsePort` is set to `USE_BLOCK_ID` . ⌋()

[SWS_Dcm_CONSTR_6068] Dependency for `DcmDspPidDataEndianness` ⌈ In case `DcmDspPidDataEndianness` is not present, the `DcmDspDataDefaultEndianness` shall be used instead. ⌋()

[SWS_Dcm_CONSTR_6069] Dependency for `DcmDspPidDataReadFnc` ⌈ `DcmDspPidDataReadFnc` shall be only present if `DcmDspPidDataUsePort` is set to `USE_DATA_SYNCH_FNC` . ⌋()

[SWS_Dcm_CONSTR_6070] Dependency for `DcmDspDataEndianness` ⌈ In case `DcmDspDataEndianness` is not present, the `DcmDspDataDefaultEndianness` shall be used instead. ⌋()

[SWS_Dcm_CONSTR_6071] Dependency for `DcmDspStartRoutineFnc` , `DcmDspStopRoutineFnc` , `DcmDspRequestRoutineResultsFnc` , `DcmDspStartRoutineConfirmationFnc` , `DcmDspStopRoutineConfirmationFnc` ⌈ The following configuration parameters shall only be present if `DcmDspRoutineUsePort` is set to FALSE.

- `DcmDspStartRoutineFnc`
- `DcmDspStopRoutineFnc`
- `DcmDspRequestRoutineResultsFnc`
- `DcmDspStartRoutineConfirmationFnc`
- `DcmDspStopRoutineConfirmationFnc`

⌋*()*

[SWS_Dcm_CONSTR_6072] Dependency for `DcmDspRoutineSignalEndianness` ⌈ In case `DcmDspRoutineSignalEndianness` is not present, the `DcmDspDataDefaultEndianness` shall be used instead. ⌋*()*

[SWS_Dcm_CONSTR_6073] Dependency for `DcmDspDataWriteFnc` ⌈ `DcmDspDataWriteFnc` shall be only present if:

- `DcmDspDataUsePort` is set to `USE_DATA_SYNCH_FNC` or
- `DcmDspDataUsePort` is set to `USE_DATA_ASYNCH_FNC` or
- `DcmDspDataUsePort` is set to `USE_DATA_ASYNCH_FNC_ERROR`

⌋*()*

[SWS_Dcm_CONSTR_6074] Dependency for `DcmDspSecurityMaxAttemptCounterReadoutTime` ⌈ `DcmDspSecurityMaxAttemptCounterReadoutTime` shall be a multiple and at minimum equal to `DcmTaskTime` . ⌋*()*

[SWS_Dcm_CONSTR_6075] Dependency for `DcmDspSecurityCompareKeyFnc` ⌈ `DcmDspSecurityCompareKeyFnc` shall be configured only if `DcmDspSecurityUsePort` is set to `USE_ASYNCH_FNC` . ⌋*()*

[SWS_Dcm_CONSTR_6076] Dependency for `DcmDspSecurityGetAttemptCounterFnc` ⌈ `DcmDspSecurityGetAttemptCounterFnc` shall be present only if `DcmDspSecurityUsePort` is set to `USE_ASYNCH_FNC` and `DcmDspSecurityAttemptCounterEnabled` is set to TRUE. ⌋*()*

[SWS_Dcm_CONSTR_6077] Dependency for `DcmDspSecurityGetSeedFnc` ⌈ `DcmDspSecurityGetSeedFnc` shall be present only if `DcmDspSecurityUsePort` is set to `USE_ASYNCH_FNC` . ⌋*()*

[SWS_Dcm_CONSTR_6078] Dependency for `DcmDspSecuritySetAttemptCounterFnc` ⌈ `DcmDspSecuritySetAttemptCounterFnc` shall be present only if `DcmDspSecurityUsePort` is set to `USE_ASYNCH_FNC` and the `DcmDspSecurityAttemptCounterEnabled` set to TRUE. ⌋*()*

[SWS_Dcm_CONSTR_6080] `DcmDspEcuResetRow` container configuration ⌈ One container `DcmDspEcuResetRow` shall be configured for each `DcmDsdSubService` (`DcmDspEcuResetId` matching to the `DcmDsdSubServiceId`) configured for the `UDS` service ECUReset (0x11) which does not have the corresponding `DcmDsdSubServiceFnc` parameter configured. ⌋*(SRS_Diag_04098)*

[SWS_Dcm_CONSTR_6081] Dependency for `DcmDspDidControlMaskBitPosition` ⌈ The value configured for `DcmDspDidControlMaskBitPosition` shall be lower than `DcmDspDidControlMaskSize` * 8. ⌋*()*

[SWS_Dcm_CONSTR_6082] Dependency for `DcmDspDidControlMaskSize` ⌈ `DcmDspDidControlMaskSize` larger than 4 shall be only allowed if `DcmDspDataUsePort` is set to `USE_DATA_ASYNCH_CLIENT_SERVER` , `USE_DATA_ASYNCH_CLIENT_SERVER_ERROR` or `USE_DATA_SYNCH_CLIENT_SERVER` . Note: ControlEnableMask larger than 32 bits is a very rare use case. Therefore the `Dcm` supports only C/S interfaces to solve this use case. ⌋*()*

[SWS_Dcm_CONSTR_6083] Dependency on `DcmDspSecurityAttemptCounterEnabled` ⌈ If `DcmDspSecurityNumAttDelay` is not configured, the `DcmDspSecurityAttemptCounterEnabled` on the same `DcmDspSecurityRow` shall be set to FALSE. ⌋*(SRS_Diag_04005)*

[SWS_Dcm_CONSTR_6084] Sender-receiver communication for IOControls is limited to atomic S/R interfaces ⌈ If a `DID` has a configured `DcmDspDidUsePort` = `USE_DATA_ELEMENT_SPECIFIC_INTERFACES` , the possible values of `DcmDspDataUsePort` are limited to non S/R interfaces. ⌋*(SRS_Diag_04218)*

[SWS_Dcm_CONSTR_6085] Atomic S/R for IOControls are limited to non-NV interfaces ⌈ If a `DID` has a configured `DcmDspDidControl` , the possible values of `DcmDspDidUsePort` are limited to atomic S/R interface and `USE_DATA_ELEMENT_SPECIFIC_INTERFACES` . ⌋*(SRS_Diag_04218)*

[SWS_Dcm_CONSTR_6086] Signals for `DID` s with Atomic S/R are not shared with other `DID` s ⌈ If a `DcmDspDid` is configured to have an atomic S/R interface, all DcmDspDataElements referenced by this `DID` shall be referenced only from this `DID` . ⌋*(SRS_Diag_04218)*

[SWS_Dcm_CONSTR_6087] Required size for white lists ⌈ If any of the optional `DcmDspAuthenticationWhiteListMemorySelectionElementRef` are configured, the corresponding `DcmDspAuthenticationWhiteListMemorySelectionMaxSize` shall be configured for that white list. ⌋*()*

[SWS_Dcm_CONSTR_6088] Supported role sizes ⌈ The parameter `DcmDspAuthenticationRoleSize` defines the size in bytes used in both, certificates and ECU internal static role configuration. All role parameters (e.g. DcmDspServiceRole) shall have values that would fit in the amount of bytes given by `DcmDspAuthenticationRoleSize` . ⌋*()*

[SWS_Dcm_CONSTR_6089] Only one compare element ⌈ In one DcmModeCondition only one of the elements DcmSwcSRDataElementRef or DcmModeConditionCertificateCompareElementRef shall be configured. ⌋*(SRS_Diag_04232)*

[SWS_Dcm_CONSTR_6090] Use of certificate compare elements ⌈ The DcmModeConditionCertificateCompareElementRef is only allowed, if the parent DcmModeRule is referenced from a DcmDspAuthenticationConnection. ⌋ *(SRS_Diag_04232)*

[SWS_Dcm_CONSTR_6091] ⌈ The presence of a DcmDsdService with DcmDsdSid TabServiceId set to 0x29, requires a configured container DcmDspAuthentication on DcmDsp. ⌋*()*

[SWS_Dcm_CONSTR_6092] ⌈ The presence of a DcmDsdService with DcmDsdSid TabServiceId set to 0x29, requires a configured DcmDspAuthenticationConnection per configured connection DcmDslConnection. ⌋*()*

[SWS_Dcm_CONSTR_6093] ⌈ Each DcmDspAuthenticationConnection shall refer a different DcmDslMainConnection by the reference in DcmDspAuthenticationConnectionMainConnectionRef. ⌋*()*

[SWS_Dcm_CONSTR_6094] ⌈ If DcmDspAuthenticationGeneralNRCModeRuleRef is configured the parameter DcmDspAuthenticationGeneralNRC shall also be configured. ⌋*()*

[SWS_Dcm_CONSTR_6095] ⌈ The presence of a DcmDsdService with DcmDsdSid TabServiceId set to 0x29, requires a DcmDsdSubService on this DcmDsdService with DcmDsdSubServiceId set to deAuthenticate. ⌋*()*

2.9 SWS_DiagnosticEventManager

[SWS_Dem_CONSTR_06118] Unique DTC values within a single event memory ⌈ The DemDtcValue shall be unique within all DTCs referencing the same event memory. ⌋*()*

[SWS_Dem_CONSTR_06119] Unique OBD DTC values within an ECU ⌈ The DemDtcValue shall be unique within all DTCs referencing the same event memory. ⌋*()*

[SWS_Dem_CONSTR_06120] Dependency for DemGeneralCallbackMonitorStatusChangedFnc ⌈ The DemGeneralCallbackMonitorStatusChangedFnc shall only be present if DemGeneralInterfaceSupport is set to TRUE." ⌋*()*

[SWS_Dem_CONSTR_06121] Dependency for DemMaxNumberEventEntryEventBuffer ⌈ The DemMaxNumberEventEntryEventBuffer shall only be present if DemEnvironmentDataCapture is set to DEM_CAPTURE_SYNCHRONOUS_TO_REPORTING (refer to DemPrimaryMemory or DemUserDefinedMemory). ⌋*()*

[SWS_Dem_CONSTR_06122] Dependency for DemOccurrenceCounterProcessing ⌈ The DemOccurrenceCounterProcessing (refer to DemPrimaryMemory or DemUserDefinedMemory) shall only be present if DemEnvironmentDataCapture is set to DEM_CAPTURE_SYNCHRONOUS_TO_REPORTING (refer to DemPrimaryMemory or DemUserDefinedMemory). ⌋*()*

[SWS_Dem_CONSTR_06123] Dependency for DemOperationCycleStatusStorage ⌈ The DemOperationCycleStatusStorage shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU . ⌋*()*

[SWS_Dem_CONSTR_06124] Dependency for DemPTOSupport ⌈ DemPTOSupport shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU . ⌋*()*

[SWS_Dem_CONSTR_06125] Dependency for DemAgingCycleCounterThreshold ⌈ DemAgingCycleCounterThreshold shall only be present if DemAgingAllowed is set to TRUE. ⌋*()*

[SWS_Dem_CONSTR_06126] Dependency for DemAgingCycleCounterThresholdForTFSLC ⌈ DemAgingCycleCounterThresholdForTFSLC shall only be present if DemStatusBitHandlingTestFailedSinceLastClear is set to DEM_STATUS_BIT_AGING_AND_DISPLACEMENT . ⌋*()*

[SWS_Dem_CONSTR_06127] Dependency for DemMaxNumberFreezeFrame Records ⌈ DemMaxNumberFreezeFrameRecords shall only be present if DemTypeOfFreezeFrameRecordNumeration is set to DEM_FF_RECNUM_CALCULATED . ⌋*()*

[SWS_Dem_CONSTR_06128] Dependency for DemAgingCycleRef ⌈ DemAgingCycleRef shall only be present if DemAgingAllowed is set to TRUE. ⌋*()*

[SWS_Dem_CONSTR_06129] Dependency for DemFreezeFrameRecNumClass Ref ⌈ DemFreezeFrameRecNumClassRef shall only be present if that DTC references a fault memory that has DemTypeOfFreezeFrameRecordNumeration is set to DEM_FF_RECNUM_CONFIGURED (refer to DemPrimaryMemory or DemUserDefinedMemory). ⌋*()*

[SWS_Dem_CONSTR_06130] Dependency for DemReportBehavior ⌈ DemReportBehavior shall only be present if DemEventKind is set to DEM_EVENT_KIND_SWC . ⌋*()*

[SWS_Dem_CONSTR_06131] Dependency for DemOBDGroupingAssociativeEventsRef ⌈ DemOBDGroupingAssociativeEventsRef shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU . ⌋ *()*

[SWS_Dem_CONSTR_06132] Dependency for DemOBDCentralizedPID21Handling ⌈ DemOBDCentralizedPID21Handling shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU . ⌋ *()*

[SWS_Dem_CONSTR_06133] Dependency for DemOBDCentralizedPID31Handling ⌈ DemOBDCentralizedPID31Handling shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU . ⌋ *()*

[SWS_Dem_CONSTR_06134] Dependency for DemOBDCompliancy ⌈ DemOBDCompliancy shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU . ⌋*()*

[SWS_Dem_CONSTR_06135] Dependency for DemOBDEngineType ⌈ DemOBDEngineType shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU . ⌋*()*

[SWS_Dem_CONSTR_06136] Dependency for DemOBDEventDisplacement ⌈ DemOBDEventDisplacement shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU . ⌋*()*

[SWS_Dem_CONSTR_06137] Dependency for DemOBDInputAcceleratorPedalInformation ⌈ DemOBDInputAcceleratorPedalInformation shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU . ⌋ *()*

[SWS_Dem_CONSTR_06138] Dependency for DemOBDInputAmbientPressure ⌈ DemOBDInputAmbientPressure shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU . ⌋*()*

[SWS_Dem_CONSTR_06139] Dependency for DemOBDInputAmbientTemperature ⌈ DemOBDInputAmbientTemperature shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU . ⌋*()*

[SWS_Dem_CONSTR_06140] Dependency for DemOBDInputDistanceInformation ⌈ DemOBDInputDistanceInformation shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU . ⌋*()*

[SWS_Dem_CONSTR_06141] Dependency for DemOBDInputEngineSpeed ⌈ DemOBDInputEngineSpeed shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU . ⌋*()*

[SWS_Dem_CONSTR_06142] Dependency for DemOBDInputEngineTemperature ⌈ DemOBDInputEngineTemperature shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU . ⌋*()*

[SWS_Dem_CONSTR_06143] Dependency for DemOBDInputProgrammingEvent ⌈ DemOBDInputProgrammingEvent shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU . ⌋*()*

[SWS_Dem_CONSTR_06144] Dependency for DemOBDInputVehicleSpeed ⌈ DemOBDInputVehicleSpeed shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU . ⌋*()*

[SWS_Dem_CONSTR_06145] Dependency for DemConsiderPtoStatus ⌈ DemConsiderPtoStatus shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU . ⌋*()*

[SWS_Dem_CONSTR_06146] Dependency for DemDtcValue ⌈ The OBD DTC DemDtcValue shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU . ⌋*()*

[SWS_Dem_CONSTR_06147] Dependency for DemEventOBDReadinessGroup ⌈ DemEventOBDReadinessGroup shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU . ⌋*()*

[SWS_Dem_CONSTR_06148] Dependency on container DemRation ⌈ The container DemRatio shall only be available if DemOBDSupport is set to DEM_OBD_MASTER_ECU . ⌋*()*

[SWS_Dem_CONSTR_06149] Dependency on container DemDtr ⌈ The container DemDtr shall only be available if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU . ⌋*()*

[SWS_Dem_CONSTR_06150] Dependency on container DemPidClass ⌈ The container DemPidClass and aggregated sub-container shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU . ⌋*()*

[SWS_Dem_CONSTR_06151] Dependency on DemCounterBasedFdcThreshold StorageValue ⌈ The configuration parameter DemCounterBasedFdcThresholdStorageValue shall only be present if DemFreezeFrameRecordTrigger is set to DEM_TRIGGER_ON_FDC_THRESHOLD or DemExtendedDataRecordTrigger is set to DEM_TRIGGER_ON_FDC_THRESHOLD or DemEventMemoryEntryStorageTrigger is set to DEM_TRIGGER_ON_FDC_THRESHOLD . ⌋*()*

[SWS_Dem_CONSTR_06152] Dependency on DemDebounceCounterJumpDown Value ⌈ DemDebounceCounterJumpDownValue shall only be present if DemDebounceCounterJumpDown is set to TRUE. ⌋*()*

[SWS_Dem_CONSTR_06153] Dependency on DemDebounceCounterJumpUp Value ⌈ DemDebounceCounterJumpUpValue shall only be present if DemDebounceCounterJumpUp is set to TRUE. ⌋*()*

[SWS_Dem_CONSTR_06154] Dependency on DemDebounceCounterStorage ⌈ DemDebounceCounterStorage shall only be present if DemOperationCycleStatusStorage is set to TRUE. ⌋*()*

[SWS_Dem_CONSTR_06155] Dependency on DemTimeBasedFdc ThresholdStorageValue ⌈ DemTimeBasedFdcThresholdStorageValue shall only be present if DemFreezeFrameRecordTrigger is set to DEM_TRIGGER_ON_FDC_THRESHOLD or DemExtendedDataRecordTrigger is set to DEM_TRIGGER_ON_FDC_THRESHOLD or DemEventMemoryEntryStorageTrigger is set to DEM_TRIGGER_ON_FDC_THRESHOLD (refer to DemPrimaryMemory or DemUserDefinedMemory). ⌋*()*

[SWS_Dem_CONSTR_06157] ⌈ Setting DemComponentFailedCallbackUsePort to TRUE is only allowed, if DemComponentFailedCallbackFnc is not configured. ⌋ *()*

[SWS_Dem_CONSTR_06158] ⌈ Existence of size parameter DemDataElementArraySize [ECUC_Dem_00949] in container DemExternalCSDataElementClass shall be present if DemDataElementDataType [ECUC_Dem_00950] in the same container is set to: UINT8_N , SINT8_N , UINT16_N , SINT16_N , UINT32_N , SINT32_N . ⌋*()*

[SWS_Dem_CONSTR_06159] ⌈ Restrictions on size parameter for 16 Bit arrays DemDataElementArraySize [ECUC_Dem_00949] shall be a multiple of 2 if the value is greater than 2 and DemDataElementDataType [ECUC_Dem_00950] is UINT16_N or SINT16_N . ⌋*()*

[SWS_Dem_CONSTR_06160] ⌈ Restrictions on size parameter for 32 Bit arrays DemDataElementArraySize [ECUC_Dem_00949] shall be a multiple of 4 if the value is greater than 4 and DemDataElementDataType [ECUC_Dem_00950] is UINT32_N or SINT32_N . ⌋*()*

[SWS_Dem_CONSTR_06161] ⌈ Existence of size parameter DemDataElementArraySize [ECUC_Dem_00967] in container DemExternalSRDataElementClass shall be present if DemDataElementDataType [ECUC_Dem_00840] in the same container is set to: UINT8_N , SINT8_N , UINT16_N , SINT16_N , UINT32_N , SINT32_N . ⌋*()*

[SWS_Dem_CONSTR_06162] ⌈ Restrictions on size parameter for 16 Bit arrays DemDataElementArraySize [ECUC_Dem_00949] shall be a multiple of 2 if the value is greater than 2 and DemDataElementDataType [ECUC_Dem_00840] is UINT16_N or SINT16_N . ⌋*()*

[SWS_Dem_CONSTR_06163] ⌈ Restrictions on size parameter for 32 Bit arrays DemDataElementArraySize [ECUC_Dem_00949] shall be a multiple of 4 if the value is greater than 4 and DemDataElementDataType [ECUC_Dem_00840] is UINT32_N or SINT32_N . ⌋*()*

[SWS_Dem_CONSTR_06165] Dependency for DemMILIndicatorRef ⌈ DemMILIndicatorRef shall only be present if DemOBDSupport is set to DEM_OBD_MASTER_ECU or DEM_OBD_PRIMARY_ECU . ⌋*()*

[SWS_Dem_CONSTR_6101] ⌈ DemExtendedDataRecordTrigger needs to be configured. DemExtendedDataRecordTrigger shall always be configured, except for internal data elements like occurrence counters. ⌋*()*

[SWS_Dem_CONSTR_6103] ⌈ In case the event combination is disabled, it is not allowed to reference from multiple events to the same dtc. ⌋*()*

[SWS_Dem_CONSTR_6104] Limitations on DemMemoryDestinationRef ⌈ If DemMirrorMemory is configured as DemMemoryDestinationRef , another DemMemoryDestinationRef on the same event of either DemPrimaryMemory or De-

mUserDefinedMemory shall be configured as a prerequisite. The same event shall not be configured two destinations if one is not DemMirrorMemory . ⌋()

[SWS_Dem_CONSTR_6106] ⌈ Only directed acyclic graph structures are supported for the dependencies of DemComponent . ⌋()

[SWS_Dem_CONSTR_6107] ⌈ Events may be assigned to exactly one DemComponent for which the monitoring is testing the error conditions. Multiple events may be assigned to the same component. ⌋()

[SWS_Dem_CONSTR_6109] ⌈ The DTC class is only available for ISO 14229-1 [1] DTCs. It is configurable per DTC optionally (refer to DemWWHOBDDTCClass). ⌋()

[SWS_Dem_CONSTR_6110] ⌈ The WWH-OBD DTC priority shall be according table **??** . ⌋()

[SWS_Dem_CONSTR_6111] ⌈ An OBD related DTC shall have an aging counter threshold of 40. ⌋()

[SWS_Dem_CONSTR_6112] ⌈ An OBD related DTC shall have the Warm-Up cycle as aging cycle. ⌋()

[SWS_Dem_CONSTR_6113] Configuration of the test failed status bit storage ⌈ For WWH-OBD ECU the DemStatusBitStorageTestFailed shall be set to True. ⌋()

[SWS_Dem_CONSTR_6114] Limitations on DemMemoryDestinationRef ⌈ A DTC can only reference the event memories via DemMemoryDestinationRef to the event memories of the same DemEventMemorySet . The scenario that a DTC references event memories via DemMemoryDestinationRef on different DemEventMemorySet is not supported. ⌋()

[SWS_Dem_CONSTR_6115] ⌈ The Dem does not support calls of

- Dem_SetEventStatus
- Dem_ResetEventStatus
- Dem_PrestoreFreezeFrame
- Dem_ClearPrestoredFreezeFrame
- Dem_ResetEventDebounceStatus

with an EventId that is referenced by any of the DemMultiEventTriggeringSlaveEventRef in container DemMultiEventTriggering . These events are exclusively used for internal triggering by calling these APIs for the master event (DemMultiEventTriggeringMasterEventRef). The behavior of the Dem is undefined if any of those APIs are called in this situation. ⌋(*SRS_Diag_04165*)

[SWS_Dem_CONSTR_6116] Limited use of monitor status change callbacks to events reported from SW-Cs only ⌈ If Dem_SetEventAvailable is called from a Cdd or BSW module, the corresponding monitor status changed callback can only be used as C-function, but not via RTE interface. ⌋()

[SWS_Dem_CONSTR_6117] ⌈ The aggregation of `DemTextTableMapping` at `DemAlternativeDataType` is only valid if the category of the CompuMethod of the DataType referenced by `DemApplicationDataType` has category set to TEXTTABLE or SCALE_LINEAR_AND_TEXTTABLE. ⌋*()*

2.10 SWS_EthernetSwitchDriver

[constr_SWS_EthSwt_CONSTR_00409] ⌈ The port specific timestamping (EthSwt PortTimeStampSupport) can be set to TRUE, if clock synchronization for connected Ethernet switches is deactivated (EthSwtClockSynchronizationSupport set to FALSE).

⌋*()*

[constr_SWS_EthSwt_CONSTR_00410] ⌈ The port specific timestamping (EthSwt PortTimeStampSupport) can be set to True, if EthSwtClockSynchronizationSupport is activated and EthSwtPortRole is not ETHSWT_UP_LINK_PORT. EthSwtPorts with EthSwtPortRole ETHSWT_UP_LINK_PORT are connected to another Ethernet switch and not considered for the time delay compensation, if EthSwtClockSynchronization Support is activated.

⌋*()*

[constr_SWS_EthSwt_CONSTR_00411] DRAFT ⌈ The ECUC partitions referenced by EthSwtConfigEcucPartitionRef shall be a subset of the ECUC partitions referenced by EthSwtEcucPartitionRef.

⌋*()*

[constr_SWS_EthSwt_CONSTR_00412] DRAFT ⌈ EthSwtConfig, EthCtrlConfig and EthTrcvConfig of one communication channel shall all reference the same ECUC partition.

⌋*()*

[constr_SWS_EthSwt_CONSTR_00413] DRAFT ⌈ The module will operate as an independent instance in each of the partitions (see ECUC_EthSwt_00129 :), means the called API will only target the partition it is called in.

⌋*()*

2.11 SWS_FunctionInhibitionManager

[SWS_Fim_CONSTR_0001] ⌈ For each configured `FiMInhibitionConfiguration` , at least one of `FiMInhSumRef` or `FiMInhEventRef` or `FiMInhComponentRef` shall be configured. ⌋*()*

2.12 SWS_GPTDriver

[constr_SWS_Gpt_CONSTR_00001] DRAFT ⌈ The ECUC partitions referenced by GptKernelEcucPartitionRef shall be a subset of the ECUC partitions referenced by Gpt EcucPartitionRef.()

⌋()

[constr_SWS_Gpt_CONSTR_00002] DRAFT ⌈ The ECUC partitions referenced by GptGroupEcucPartitionRef shall be a subset of the ECUC partitions referenced by Gpt EcucPartitionRef.

⌋()

2.13 SWS_ICUDriver

[constr_SWS_Icu_CONSTR_00001] DRAFT ⌈ The ECUC partitions referenced by IcuKernelEcucPartitionRef shall be a subset of the ECUC partitions referenced by Icu EcucPartitionRef.

⌋()

[constr_SWS_Icu_CONSTR_00002] DRAFT ⌈ The ECUC partitions referenced by IcuChannelEcucPartitionRef shall be a subset of the ECUC partitions referenced by IcuEcucPartitionRef.

⌋()

2.14 SWS_LINDriver

[constr_SWS_Lin_CONSTR_00270] DRAFT ⌈ The module will operate as an independent instance in each of the partitions, means the called API will only target the partition it is called in.

⌋()

[constr_SWS_Lin_CONSTR_00278] DRAFT ⌈ The ECUC partitions referenced by LinChannelEcucPartitionRef shall be a subset of the ECUC partitions referenced by LinEcucPartitionRef.

⌋()

[constr_SWS_Lin_CONSTR_00279] DRAFT ⌈ LinChannel and LinTrcvChannel of one communication channel shall all reference the same ECUC partition.

⌋()

2.15 SWS_MCUDriver

[constr_SWS_Mcu_CONSTR_00001] : DRAFT ⌈ The module will operate as an independent instance in each of the partitions, means the called API will only target the partition it is called in.

⌋*()*

2.16 SWS_PWMDriver

[constr_SWS_Pwm_CONSTR_00001] DRAFT ⌈ The ECUC partitions referenced by PwmKernelEcucPartitionRef shall be a subset of the ECUC partitions referenced by PwmEcucPartitionRef.

⌋*()*

2.17 SWS_PortDriver

[constr_SWS_Port_CONSTR_00233] DRAFT ⌈ The ECUC partitions referenced by PortPinEcucPartitionRef shall be a subset of the ECUC partitions referenced by Port EcucPartitionRef.

⌋*()*

2.18 SWS_RTE

[SWS_Rte_CONSTR_03510] Exclude usage of `OS_SPINLOCK` in `RteExclusiveAreaImplementation` ⌈ The usage of the enumeration literal `OS_SPINLOCK` for the parameter `RteExclusiveAreaImplMechanism` shall be excluded if the parameter `RteExclusiveAreaImplMechanism` is used in the context of the container `RteExclusiveAreaImplementation` . ⌋*()*

[SWS_Rte_CONSTR_03870] ⌈ In case that `RteDevErrorDetectUninit` is configured to true, `RteDevErrorDetect` shall be configured to true. ⌋*()*

[SWS_Rte_CONSTR_03873] ⌈ All `OperationInvokedEvent` s/ `BswOperationInvokedEvent` s which are activating the same `server ExecutableEntity` shall be mapped by at most one `RteEventToTaskMapping` / `RteBswEventToTaskMapping` which references an `OsTask` . ⌋*(SRS_Rte_00019, SRS_Rte_00033)*

[SWS_Rte_CONSTR_03874] ⌈ A `RteEventToTaskMapping` / `RteBswEventToTaskMapping` shall only own more than one `RteEventRef` / `RteBswEventRef` reference if all owned `RteEventRef` s/ `RteBswEventRef` s refer to `OperationInvokedEvent` s/ `BswOperationInvokedEvent` s which in turn are triggering the same `server ExecutableEntity` . ⌋*(SRS_Rte_00019, SRS_Rte_00033)*

[SWS_Rte_CONSTR_04558] ⌈ An OsTask shall be part of at most one task chain. Hence, an OsTask shall be referenced by at most one RtePredecessorOsTaskRef and by at most one RteSuccessorOsTaskRef . ⌋*(SRS_Rte_00049)*

[SWS_Rte_CONSTR_04559] ⌈ The configuration of RteOsTaskChain s shall not define circular chains. ⌋*(SRS_Rte_00049)*

[SWS_Rte_CONSTR_09000] Rte_IFeedback API may only be used by the RunnableEntity s that describe its usage ⌈ The Rte_IFeedback API shall only be used by a RunnableEntity that either has a VariableAccess in the dataWriteAccess role referring to the VariableDataPrototype or is triggered by a DataWriteCompletedEvent referring to the VariableAccess which in turn references the VariableDataPrototype . ⌋*()*

[SWS_Rte_CONSTR_09001] Whole DataPrototypeGroup in role dpgRequiresCoherency shall be propagated coherently ⌈ All RunnableEntity s in a RunnableEntityGroup with dataWriteAccess to data belonging to the same DataPrototypeGroup in the role dpgRequiresCoherency shall

- Be mapped to the same OS Task

  AND shall

- A) either be scheduled in a way that these RunnableEntity s can not be interrupted by RunnableEntity s with dataReadAccess to (more than one) data belonging to the DataPrototypeGroup .
- B) or the RteImplicitCommunication shall be configured to ensure a coherent propagation (RteCoherentAccess == true) for reading RunnableEntity s [1] .

⌋*()*

[SWS_Rte_CONSTR_09002] The whole DataPrototypeGroup shall be read stable for the whole RunnableEntityGroup in the role regRequiresStability ⌈ .

All RunnableEntity s with dataReadAccess to data belonging to the same DataPrototypeGroup and which are belonging to the same RunnableEntityGroup in the role regRequiresStability shall

- either be configured in a way that the chain of RunnableEntity s with dataReadAccess to the data of the DataPrototypeGroup can not be interrupted by any of the RunnableEntity (s) with dataWriteAccess to data of the DataPrototypeGroup
- or the RteImplicitCommunication shall be configured to ensure stable data values (RteCoherentAccess == true) for reading RunnableEntity s belonging to the RunnableEntityGroup .

[1] RunnableEntity s with have as well dataWriteAccess to data belonging to the DataPrototypeGroup are excluded because inside the calculation chain the latest data values are visible

⌋()

[SWS_Rte_CONSTR_09005] The references `RteSwcTriggerSourceRef` has to be consistent with the `RteSoftwareComponentInstanceRef` ⌈ The references `RteSwcTriggerSourceRef` has to be consistent with the `RteSoftwareComponentInstanceRef` . This means the referenced `Trigger` / `InternalTriggeringPoint` has to belong to the `AtomicSwComponentType` which is referenced by the related `SwComponentPrototype` . ⌋()

[SWS_Rte_CONSTR_09006] The references `RteBswTriggerSourceRef` has to be consistent with the `RteBswImplementationRef` ⌈ The references `RteBswTriggerSourceRef` has to be consistent with the `RteBswImplementationRef` . This means the referenced `Trigger` / `BswInternalTriggeringPoint` has to belong to the `BswModuleDescription` which is referenced by the related `BswImplementation` . ⌋()

[SWS_Rte_CONSTR_09007] *issuedTrigger* and *BswTriggerDirectImplementation* are mutually exclusive ⌈ A *releasedTrigger* *Trigger* shall not be referenced by both a *issuedTrigger* and a *BswTriggerDirectImplementation* . ⌋()

[SWS_Rte_CONSTR_09008] The same `Trigger` in a *trigger sink* must not be connected to multiple *trigger source* s ⌈ The same `Trigger` in a *trigger sink* must not be connected to multiple *trigger source* s. ⌋()

[SWS_Rte_CONSTR_09009] Synchronized `Trigger` shall not be referenced by more than one type of access method ⌈ A synchronized `Trigger` shall only be referenced by either `ExternalTriggeringPoint` s, `issuedTrigger` s or `BswTriggerDirectImplementation` s. ⌋()

[SWS_Rte_CONSTR_09010] Worst case execution time shall be less than the GCD ⌈ The `RunnableEntity` s or `BswSchedulableEntity` s worst case execution time shall be less than the GCD of all `BswSchedulableEntity` s and `RunnableEntity` s period and offset in activation offset context for `RunnableEntity` s and `BswSchedulableEntity` s. ⌋()

[SWS_Rte_CONSTR_09011] `NvMBlockDescriptor` related to a `RAM Block` of a `NvBlockSwComponentType` shall use `NvmBlockUseSyncMechanism` ⌈ The `NVRAM Block` associated to the `NvBlockDescriptor` s of a `NvBlockSwComponentType` shall be configured with the `NvMBlockUseSyncMechanism` feature enabled, and the `NvMWriteRamBlockToNvCallback` and `NvMReadRamBlockFromNvCallback` parameters set to the `Rte_GetMirror` and `Rte_SetMirror` API of the `NvBlockDescriptor` . ⌋()

[SWS_Rte_CONSTR_09012] Category 1 interrupts shall not access the RTE. ⌈ Category 1 interrupts shall not access the RTE. ⌋()

[SWS_Rte_CONSTR_09013] Exactly one mode or one mode transition shall be active ⌈ Whenever any `RunnableEntity` or `BswSchedulableEntity` is running, there shall always be exactly one mode or one mode transition active of each `ModeDeclarationGroupPrototype` . ⌋()

[SWS_Rte_CONSTR_09014] *ModeSwitchPoint* (s) and *managedModeGroup* (s) are mutually exclusive for synchronized *ModeDeclarationGroupPrototype* s ⌈ Only one of two synchronized *ModeDeclarationGroupPrototype* s shall mutual exclusively be referenced by *ModeSwitchPoint* (s) or *managedModeGroup* association(s). ⌋ *()*

[SWS_Rte_CONSTR_09015] `Rte_Write` API may only be used by the runnable that describe its usage ⌈ The `Rte_Write` API may only be used by the runnable that contains the corresponding `VariableAccess` in the `dataSendPoint` role ⌋*()*

[SWS_Rte_CONSTR_09016] `Rte_Send` API may only be used by the runnable that describes its usage ⌈ The `Rte_Send` API may only be used by the runnable that contains the corresponding `VariableAccess` in the `dataSendPoint` role ⌋*()*

[SWS_Rte_CONSTR_09017] `Rte_Switch` API may only be used by the runnable that describes its usage ⌈ The `Rte_Switch` API may only be used by the runnable that contains the corresponding *ModeSwitchPoint* ⌋*()*

[SWS_Rte_CONSTR_09018] `Rte_Invalidate` API may only be used by the runnable that describe its usage ⌈ The `Rte_Invalidate` API may only be used by the runnable that contains the corresponding `VariableAccess` in the `dataSendPoint` role ⌋*()*

[SWS_Rte_CONSTR_09019] `Rte_Feedback` API may only be used by the runnable that describe its usage ⌈ A blocking `Rte_Feedback` API may only be used by the runnable that contains the corresponding `WaitPoint` ⌋*()*

[SWS_Rte_CONSTR_09020] The blocking `Rte_SwitchAck` API may only be used by the runnable that describes its usage. ⌈ A blocking `Rte_SwitchAck` API must only be used by the runnable that contains the corresponding `WaitPoint` ⌋ *()*

[SWS_Rte_CONSTR_09021] `Rte_Read` API may only be used by the runnable that describe its usage ⌈ The `Rte_Read` API may only be used by the runnable that contains the corresponding `VariableAccess` in the `dataReceivePointByArgument` role ⌋*()*

[SWS_Rte_CONSTR_09022] `Rte_DRead` API may only be used by the runnable that describe its usage ⌈ The `Rte_DRead` API may only be used by the runnable that contains the corresponding `VariableAccess` in the `dataReceivePointByValue` role ⌋*()*

[SWS_Rte_CONSTR_09023] `Rte_Receive` API may only be used by the runnable that describe its usage ⌈ The `Rte_Receive` API may only be used by the runnable that contains the corresponding `VariableAccess` in the `dataReceivePointByArgument` role ⌋*()*

[SWS_Rte_CONSTR_09024] `Rte_Call` API may only be used by the runnable that describe its usage ⌈ The `Rte_Call` API may only be used by the runnable that contains the corresponding `ServerCallPoint` ⌋*()*

[SWS_Rte_CONSTR_09025] Blocking `Rte_Result` API may only be used by the runnable that describe the `WaitPoint` ⌈ The blocking `Rte_Result` API may only be used by the runnable that contains the corresponding `WaitPoint` ⌋*()*

[SWS_Rte_CONSTR_09026] `Rte_IWriteRef` may not return values written in previous executions ⌈ The reference returned by `Rte_IWriteRef` shall not be used by the runnables for reading the value previously written. ⌋*()*

[SWS_Rte_CONSTR_09027] `Rte_IStatus` API shall only be used by a `RunnableEntity` describing an read access to the related data ⌈ The `Rte_IStatus` API shall only be used by a `RunnableEntity` that has a `VariableAccess` in the `dataReadAccess` role referring to the `VariableDataPrototype` to which the status belongs. ⌋*()*

[SWS_Rte_CONSTR_09028] `Rte_Enter` and `Rte_Exit` API may only be used by runnables describing its usage ⌈ The `Rte_Enter` and `Rte_Exit` API may only be used by *Runnable Entities* that contain a corresponding *canEnterExclusiveArea* association ⌋*()*

[SWS_Rte_CONSTR_09029] Nested call of `Rte_Enter` and `Rte_Exit` is restricted ⌈ The `Rte_Enter` and `Rte_Exit` API may only be called nested if different exclusive areas are invoked; in this case exclusive areas shall exited in the reverse order they were entered. ⌋*()*

[SWS_Rte_CONSTR_09030] `Rte_Mode` API may only be used by the runnable that describe its usage ⌈ The `Rte_Mode` API may only be used by the runnable that contains the corresponding `ModeAccessPoint` ⌋*()*

[SWS_Rte_CONSTR_09031] `Rte_Mode` API may only be used by the runnable that describe its usage ⌈ The `Rte_Mode` API may only be used by the runnable that contains the corresponding `ModeAccessPoint` ⌋*()*

[SWS_Rte_CONSTR_09032] `Rte_Trigger` API may only be used by the runnable that describe its usage ⌈ The `Rte_Trigger` API may only be used by the runnable that contains the corresponding `ExternalTriggeringPoint` . ⌋*()*

[SWS_Rte_CONSTR_09033] `Rte_IrTrigger` API may only be used by the runnable that describe its usage ⌈ The `Rte_IrTrigger` API may only be used by the runnable that contains the corresponding `InternalTriggeringPoint` . ⌋*()*

[SWS_Rte_CONSTR_09034] `Rte_IsUpdated` API may only be used by the runnable that describe the access to the corresponding data ⌈ The `Rte_IsUpdated` API may only be used by the runnable that contains the corresponding `VariableAccess` in the `dataReceivePointByArgument` or `dataReceivePointByValue` role. ⌋*()*

[SWS_Rte_CONSTR_09035] `Rte_Start` shall be called only once ⌈ `Rte_Start` shall be called only once by the EcuStateManager from trusted OS context on a core after the basic software modules required by RTE are initialized. ⌋*()*

[SWS_Rte_CONSTR_09036] `Rte_Start` API may only be used after call of `SchM_Init` ⌈ The `Rte_Start` API may only be used after the *Basic Software Scheduler* is initialized (after termination of the `SchM_Init`). ⌋*()*

[SWS_Rte_CONSTR_09037] `Rte_Start` API shall be called on every core ⌈ The `Rte_Start` API shall be called on every core that hosts AUTOSAR software-components of the ECU. ⌋*()*

[SWS_Rte_CONSTR_09038] `Rte_Stop` shall be called before BSW shutdown ⌈ `Rte_Stop` shall be called by the EcuStateManager before the basic software modules required by RTE are shut down. ⌋*()*

[SWS_Rte_CONSTR_09039] `Rte_PartitionTerminated` shall be called only once ⌈ `Rte_PartitionTerminated` shall be called only once by the Protection Hook. ⌋*()*

[SWS_Rte_CONSTR_09040] `Rte_PartitionRestarting` shall be called only onc ⌈ `Rte_PartitionRestarting` shall be called only once by the ProtectionHook. ⌋*()*

[SWS_Rte_CONSTR_09041] `Rte_RestartPartition` shall be called from `RestartTask` ⌈ `Rte_RestartPartition` shall be called only in the context of the `RestartTask` of the given partition. ⌋*()*

[SWS_Rte_CONSTR_09042] `Array Implementation Data Types` needs at least one element ⌈ The `arraySize` defining number of elements in one dimension of an *Array Implementation Data Type* shall be an integer that is ≥ 1 for each dimension. ⌋*()*

[SWS_Rte_CONSTR_09043] `Structure Implementation Data Types` needs at least one element ⌈ A structure shall include at least one element defined by a `ImplementationDataTypeElement` . ⌋*()*

[SWS_Rte_CONSTR_09045] The upper two bits of the of the server return value are reserved ⌈ Only the least significant six bit of the return value of a server runnable shall be used by the application to indicate an error. The upper two bit shall be zero. ⌋ *()*

[SWS_Rte_CONSTR_09046] `SchM_Enter` and `SchM_Exit` API may only be used by `BswModuleEntity` s describing its usage ⌈ The `SchM_Enter` and `SchM_Exit` API may only be used by `BswModuleEntity` s that contain a corresponding `canEnterExclusiveArea` association ⌋*()*

[SWS_Rte_CONSTR_09047] Nested call of `SchM_Enter` and `SchM_Exit` API is restricted ⌈ The `SchM_Enter` and `SchM_Exit` API may only be called nested if different exclusive areas are invoked; in this case exclusive areas shall exited in the reverse order they were entered. ⌋*()*

[SWS_Rte_CONSTR_09048] `SchM_Exit` API may only be used by `BswModuleEntity` s that describe its usage ⌈ The `SchM_Exit` API may only be used

by BswModuleEntity s that contain a corresponding canEnterExclusiveArea association ⌋()

[SWS_Rte_CONSTR_09049] SchM_Switch API may only be used by BswModuleEntity s that describe its usage ⌈ The SchM_Switch API may only be used by BswModuleEntity s that contain a corresponding managedModeGroup association ⌋()

[SWS_Rte_CONSTR_09050] SchM_Mode API may only be used by BswModuleEntity s that describe its usage ⌈ The SchM_Mode API may only be used by BswModuleEntity s that contain a corresponding managedModeGroup association or accessedModeGroup association ⌋()

[SWS_Rte_CONSTR_09051] SchM_Mode API may only be used by BswModuleEntity s that describe its usage ⌈ The SchM_Mode API may only be used by BswModuleEntity s that contain a corresponding managedModeGroup association or accessedModeGroup association ⌋()

[SWS_Rte_CONSTR_09052] SchM_SwitchAck API may only be used by BswModuleEntity s that describe its usage ⌈ The SchM_SwitchAck API may only be used by BswModuleEntity s that contain a corresponding managedModeGroup association ⌋()

[SWS_Rte_CONSTR_09053] SchM_Trigger API may only be used by the BswModuleEntity s that describe its usage ⌈ The SchM_Trigger API may only be used by the BswModuleEntity that contains the corresponding issuedTrigger association. ⌋()

[SWS_Rte_CONSTR_09054] SchM_ActMainFunction API may only be used by the BswModuleEntity s that describe its usage ⌈ The SchM_ActMainFunction API may only be used by the BswModuleEntity that contains the corresponding activationPoint association. ⌋()

[SWS_Rte_CONSTR_09055] SchM_Init , SchM_Start , SchM_StartTiming shall be called only once ⌈ SchM_Init , SchM_Start , SchM_StartTiming shall be called only once by the *EcuStateManager* on each core after the basic software modules required by the *Basic Software Scheduler* part of the RTE are initialized. ⌋()

[SWS_Rte_CONSTR_09056] SchM_Deinit API may only be used after the was RTE finalized ⌈ The SchM_Deinit API may only be used after the RTE finalized (after termination of the Rte_Stop) ⌋()

[SWS_Rte_CONSTR_09057] SchM_Deinit shall be called before shut down of BSW ⌈ SchM_Deinit shall be called by the *EcuStateManager* before the basic software modules required by *Basic Software Scheduler* part are shut down. ⌋()

[SWS_Rte_CONSTR_09058] BswSchedulableEntity is not allowed to have service arguments or return value ⌈ The Basic Software Scheduler requires that the BswModuleEntry has no service arguments (unless SchM_ActivatingEvent is enabled) and no return value. ⌋()

[SWS_Rte_CONSTR_09059] Usage of *Basic Software Scheduler* API prerequisites the include of the *Module Interlink Header File* ⌈ Each BSW module implementation shall include its *Module Interlink Header File* if it uses *Basic Software Scheduler* API or if it implements BswSchedulableEntity s. ⌋()

[SWS_Rte_CONSTR_09060] `Rte_Init` API may only be used after call of `Rte_Start` ⌈ The Rte_Init API may only be used after the *RTE* is initialized (after termination of the Rte_Start). ⌋()

[SWS_Rte_CONSTR_09061] `Rte_StartTiming` API may only be used after call of `Rte_Start` ⌈ The Rte_StartTiming API may only be used after the *RTE* is initialized (after termination of the Rte_Start). ⌋()

[SWS_Rte_CONSTR_09062] Entire mapping of `on-entry Runnable Entities` for `initialMode` to `RteInitializationRunnableBatch` containers ⌈ Either all or none of the on-entry Runnable Entities of a particular mode machine instance for the initialMode shall be mapped to RteInitializationRunnableBatch containers. ⌋()

[SWS_Rte_CONSTR_09063] Restricted kinds of `RTEEvent` s which may mapped to `RteInitializationRunnableBatch` containers ⌈ Only SwcModeSwitchEvent s with activation = onEntry and referring to the initialMode or InitEvent s may be mapped to RteInitializationRunnableBatch containers with the means of a RteUsedInitFnc reference. ⌋()

[SWS_Rte_CONSTR_09064] A single `RteInitializationRunnableBatch` container may not handle `RTEEvent` s of different partitions ⌈ All RTEEvent s mapped to a RteInitializationRunnableBatch container may only trigger RunnableEntity s belonging to the same partition. ⌋()

[SWS_Rte_CONSTR_09076] SchM_Result API may only be used by the Bsw ModuleEntity that describe its usage ⌈ The SchM_Result API may only be used within the BswModuleEntity that references the corresponding BswAsynchronousServerCallResultPoint using a callPoint association. ⌋()

[SWS_Rte_CONSTR_09077] SchM_Send API may only be used by the Bsw ModuleEntity that describes its usage ⌈ The SchM_Send API may only be used within the BswModuleEntity that references the VariableDataPrototype using a dataSendPoint . ⌋()

[SWS_Rte_CONSTR_09078] SchM_Receive API may only be used by the Bsw ModuleEntity that describes its usage ⌈ The SchM_Receive API may only be used within the BswModuleEntity that references the VariableDataPrototype using a dataReceivePoint . ⌋()

[SWS_Rte_CONSTR_09079] SchM_Call API may only be used by the BswModule Entity that describe its usage ⌈ The SchM_Call API may only be used within the BswModuleEntity that references the corresponding BswSynchronousServerCallPoint respectively BswAsynchronousServerCallPoint using a callPoint association. ⌋()

[SWS_Rte_CONSTR_09080] The *shortNames* of *PortInterfaces* shall be unique within a software component if it supports multiple instantiation or `indirectAPI` attribute is set to 'true' ⌈ The *shortNames* of *PortInterfaces* shall be unique within a software component for each set of PPortPrototypes or RPortPrototypes if the software component supports multiple instantiation or if the `indirectAPI` attribute is set to 'true' for at least one require or provide port.

This is required to generate distinguishable Port Data Structure data types. ⌋*()*

[SWS_Rte_CONSTR_09081] Mapping to partition vs the value of VariableAccess.scope ⌈ For every connection between `SwComponentPrototype` s mapped to different partitions the value of `VariableAccess` . `scope` shall not be set to VariableAccessScopeEnum.communicationIntraPartition. ⌋*()*

[SWS_Rte_CONSTR_09082] `RtePositionInTask` and `RteBswPositionInTask` values shall be unique in a particular context ⌈ `RtePositionInTask` and `RteBswPositionInTask` shall have unique values for any particular task in the case `RTEEvent` s and `BswEvent` s are mapped to `OsTask` s and shall have unique values for any particular scope of direct invocation in the case that the a direct function call is configured. ⌋*()*

[SWS_Rte_CONSTR_09083] `Rte_IRead` API may only be used by the runnable that describe its usage ⌈ The `Rte_IRead` API may only be used by the runnable that contains the corresponding `VariableAccess` in the `dataReadAccess` role. ⌋*()*

[SWS_Rte_CONSTR_09084] `Rte_IWrite` API may only be used by the runnable that describe its usage ⌈ The `Rte_IWrite` API may only be used by the runnable that contains the corresponding `VariableAccess` in the `dataWriteAccess` role. ⌋*()*

[SWS_Rte_CONSTR_09085] `Rte_IWriteRef` API may only be used by the runnable that describe its usage ⌈ The `Rte_IWriteRef` API may only be used by the runnable that contains the corresponding `VariableAccess` in the `dataWriteAccess` role. ⌋*()*

[SWS_Rte_CONSTR_09086] `Rte_IInvalidate` API may only be used by the runnable that is describing an write access to the data ⌈ The `Rte_IInvalidate` API may only be used by the runnable that contains the corresponding `VariableAccess` in the `dataWriteAccess` role to the `VariableDataPrototype` where the associated `InvalidationPolicy` of the `VariableDataPrototype` is set to `keep` or `replace` . ⌋*()*

[SWS_Rte_CONSTR_09087] `Rte_IrvIRead` API may only be used by the runnable that describe its usage ⌈ The `Rte_IrvIRead` API may only be used by the runnable that contains the corresponding `VariableAccess` in the `readLocalVariable` role. ⌋*()*

[SWS_Rte_CONSTR_09088] `Rte_IrvIWrite` API may only be used by the runnable that describe its usage ⌈ The `Rte_IrvIWrite` API may only be used

by the runnable that contains the corresponding VariableAccess in the writtenLocalVariable role. ⌋()

[SWS_Rte_CONSTR_09089] Rte_IrvRead API may only be used by the runnable that describe its usage ⌈ The Rte_IrvRead API may only be used by the runnable that contains the corresponding VariableAccess in the readLocalVariable role. ⌋()

[SWS_Rte_CONSTR_09090] Rte_IrvWrite API may only be used by the runnable that describe its usage ⌈ The Rte_IrvWrite API may only be used by the runnable that contains the corresponding VariableAccess in the writtenLocalVariable role. ⌋()

[SWS_Rte_CONSTR_09091] RteSwNvRamMappingRef and RteSwNvBlockDescriptorRef are excluding each other ⌈ If an RteSwNvBlockDescriptorRef is defined there shall be no RteSwNvRamMappingRef , RteNvmRomBlockLocationSymbol and RteNvmRamBlockLocationSymbol defined. If an RteSwNvRamMappingRef is defined there shall be no RteSwNvBlockDescriptorRef defined. ⌋()

[SWS_Rte_CONSTR_09092] Rte_IrvIWriteRef API may only be used by the runnable that describe its usage ⌈ The Rte_IrvIWriteRef API may only be used by the runnable that contains the corresponding VariableAccess in the writtenLocalVariable role. ⌋()

[SWS_Rte_CONSTR_09093] Rte_IrvIWriteRef may not return values written in previous executions ⌈ The reference returned by Rte_IrvIWriteRef shall not be used by the runnables for reading the value previously written. ⌋()

[SWS_Rte_CONSTR_09100] The reference RteSwcModeManagerRef has to be consistent with the RteSoftwareComponentInstanceRef ⌈ The reference RteSwcModeManagerRef has to be consistent with the RteSoftwareComponentInstanceRef . This means the referenced ModeDeclarationGroupPrototype shall be instantiated in the context of an AbstractProvidedPortPrototype owned by the AtomicSwComponentType which is referenced by the related SwComponentPrototype . ⌋()

[SWS_Rte_CONSTR_09101] The reference RteBswModeManagerRef has to be consistent with the RteBswImplementationRef ⌈ The reference RteBswModeManagerRef has to be consistent with the RteBswImplementationRef . This means the referenced ModeDeclarationGroupPrototype has to be a providedModeGroup in the BswModuleDescription which is referenced by the related BswImplementation . ⌋()

[SWS_Rte_CONSTR_09102] Exclusive usage of OsTask s used for distributed shared mode queue ⌈ An OsTask belonging to a distributed shared mode queue shall have only mapped on-entry ExecutableEntitys , on-transition ExecutableEntitys , on-exit ExecutableEntitys , and ModeSwitchAck ExecutableEntitys to it which are triggered by mode machine instance s belonging to the identical distributed shared mode queue . ⌋()

[SWS_Rte_CONSTR_80000] DRAFT ⌈ `RTE_PLUGIN` in `RteExclusiveAreaImplementation` requires the configuration of an `RTE Implementation Plug-In`

The usage of the enumeration literal `RTE_PLUGIN` for the parameter `RteExclusiveAreaImplMechanism` requires the configuration of the reference `RteExclusiveAreaResponsibleRipsPluginRef` in the owning container `RteExclusiveAreaImplementation`. ⌋*(SRS_Rte_00302)*

[SWS_Rte_CONSTR_80001] DRAFT ⌈ `RTE_PLUGIN` in `RteBswExclusiveAreaImpl` requires the configuration of an `RTE Implementation Plug-In`

The usage of the enumeration literal `RTE_PLUGIN` for the parameter `RteExclusiveAreaImplMechanism` requires the configuration of the reference `RteBswExclusiveAreaResponsibleRipsPluginRef` in the owning container `RteBswExclusiveAreaImpl`. ⌋*(SRS_Rte_00302)*

[SWS_Rte_CONSTR_80002] DRAFT ⌈ Valid instance reference targets of `Rte Implementation Plug-Ins`

The `RIPS FlatInstanceDescriptors` for a `Data Communication Graph` shall reference the data instances according table 2.1 ⌋*(SRS_Rte_00300, SRS_Rte_00301)*

Data Communication Graph involves NvBlockSwComponent	*Conversion*	*Communication multiplicity*	*RIPS FlatInstanceDescriptors*
No	No	1:n	`VariableDataPrototype` instance in the `AbstractProvidedPortPrototype`
No	No	n:1	`VariableDataPrototype` instance in the `RPortPrototype`
No	No	n:m where n > 1 and m > 1	`VariableDataPrototype` instance in any of the `PRPortPrototype` s
Yes	No	n:m where n >= 1 and m >= 1	`VariableDataPrototype` instance in the `AbstractProvidedPortPrototype` at the `NvBlockSwComponent`
No	Yes	1:n	`VariableDataPrototype` instance in the `AbstractProvidedPortPrototype` AND one per different representation of `VariableDataPrototype` instance in the `RPortPrototype`
No	Yes	n:1	`VariableDataPrototype` instance in the `AbstractRequiredPortPrototype` AND one per different representation of `VariableDataPrototype` instance in the `PPortPrototype`

▽

△

Yes	Yes	where n >= 1 and m >= 1	VariableDataPrototype instance in the AbstractProvidedPortPrototype at the NvBlockSwComponent AND one per different representation of VariableDataPrototype instance in the PortPrototype

Table 2.1: Reference targets of RIPS FlatInstanceDescriptors

[SWS_Rte_CONSTR_80003] DRAFT ⌈ A Data Communication Graph is handled by at most one RTE Implementation Plug-In

In the case that a Data Communication Graph is referenced by several RIPS FlatInstanceDescriptors all those RIPS FlatInstanceDescriptors shall reference via FlatInstanceDescriptor . rtePluginProps . associatedRtePlugin the identical RteRipsPluginProps container. ⌋*(SRS_Rte_00300, SRS_Rte_00301)*

[SWS_Rte_CONSTR_80004] DRAFT ⌈ A Client Server Communication Graph is handled by at most one RTE Implementation Plug-In

In the case that a Client Server Communication Graph is referenced by several RIPS FlatInstanceDescriptors , all those RIPS FlatInstanceDescriptors shall reference via FlatInstanceDescriptor . rtePluginProps . associatedRtePlugin the same RteRipsPluginProps container. ⌋ *(SRS_Rte_00312)*

[SWS_Rte_CONSTR_80005] DRAFT ⌈ Valid operation instance reference for Rte Implementation Plug-Ins I

The RIPS FlatInstanceDescriptor for a Client Server Communication Graph shall reference the operation instance in the AbstractProvidedPortPrototype , if the configuration contains only the Server or the Clients and Server for the Client Server Communication Graph . ⌋*(SRS_Rte_00312)*

[SWS_Rte_CONSTR_80006] DRAFT ⌈ Valid operation instance reference for Rte Implementation Plug-Ins II

The RIPS FlatInstanceDescriptor for a Client Server Communication Graph shall reference the operation instance in the RPortPrototype , if the configuration contains only the Clients for the Client Server Communication Graph . ⌋*(SRS_Rte_00312)*

[SWS_Rte_CONSTR_80007] DRAFT ⌈ Valid operation instance reference for Rte Implementation Plug-Ins III

The RIPS FlatInstanceDescriptor for a Client Server Communication Graph is only applicable, if the client server communication configures a transformer according [SWS_Rte_08794] (inter ECU) or via ClientServerOperationMapping (intra ECU). ⌋*(SRS_Rte_00312)*

[SWS_Rte_CONSTR_80009] DRAFT ⌈ Mandatory `Rte_Rips_InvocationHandler` in case of transformers

In the case a `server runnable` or `triggered runnable` invoked by an `RTE Implementation Plug-In` handles the transformers the belonging `RteEvent-ToTaskMapping` shall define an `RteRipsInvocationHandlerRef` . ⌋ *(SRS_Rte_00312, SRS_Rte_00317)*

[SWS_Rte_CONSTR_80010] DRAFT ⌈ Partitions shall have the same life-cycle

All partitions affecting the same `RTE Implementation Plug-In` shall have the same life-cycle. ⌋*(SRS_Rte_00307, SRS_Rte_00309)*

[SWS_Rte_CONSTR_80011] DRAFT ⌈ Limitation on `RTE Implementation Plug-In` support for `rptPreparationLevel` s

`Data Communication Graph` s with `rptPreparationLevel` s greater than `rptLevel1` shall not be assigned to an `RTE Implementation Plug-In` . ⌋ *(SRS_Rte_00244)*

[SWS_Rte_CONSTR_80012] DRAFT ⌈ `mode machine instance` belonging to a `distributed shared mode queue` is not allowed to be configured for individual `RTE Implementation Plug-In` support

In case a `mode machine instance` belongs to a `distributed shared mode queue` the reference `RteModeMachineInstanceResponsibleRipsPluginRef` shall not be configured. ⌋*(SRS_Rte_00310)*

[SWS_Rte_CONSTR_80013] DRAFT ⌈ Restrictions on direct function call configurations in the scope of `RTE Implementation Plug-Ins`

If an RTE Generator supports an activation of `ExecutableEntity` s via direct function call listed in [SWS_Rte_80029] only when the support for `RTE Implementation Plug-Ins` is enabled the input configuration needs to fulfill following condition:

- all `Communication Graphs` , `ExclusiveArea` s and `mode machine instances` accessed by the to-be-activated `ExecutableEntity` are assigned to `RTE Implementation Plug-Ins` s

  AND

- the to-be-activated `ExecutableEntity` do not in turn activate `RTEEvent` s or `BswEvent` s which are mapped to `OsTask` s.

⌋*(SRS_Rte_00305)*

[SWS_Rte_CONSTR_80014] DRAFT ⌈ A `Trigger Communication Graph` is handled by at most one `RTE Implementation Plug-In`

In the case that a `Trigger Communication Graph` is referenced by several `RIPS FlatInstanceDescriptors` , all those `RIPS FlatInstanceDescriptors` shall reference via `FlatInstanceDescriptor` . `rtePluginProps` . `associatedRtePlugin` the same `RteRipsPluginProps` container. ⌋*(SRS_Rte_00317)*

[SWS_Rte_CONSTR_80015] DRAFT ⌈ Valid `trigger` instance reference for `Rte Implementation Plug-Ins` I

The `RIPS FlatInstanceDescriptor` for a `Trigger Communication Graph` shall reference the `trigger` instance in the `AbstractProvidedPortPrototype` , if the configuration contains only the trigger source or the trigger sink(s) and trigger source for the `Trigger Communication Graph` . ⌋*(SRS_Rte_00317)*

[SWS_Rte_CONSTR_80016] DRAFT ⌈ Valid `trigger` instance reference for `Rte Implementation Plug-Ins` II

The `RIPS FlatInstanceDescriptor` for a `Trigger Communication Graph` shall reference the `trigger` instance in the `RPortPrototype` , if the configuration contains only the trigger sink for the `Trigger Communication Graph` . ⌋ *(SRS_Rte_00317)*

[SWS_Rte_CONSTR_80017] DRAFT ⌈ Valid `trigger` instance reference for `Rte Implementation Plug-Ins` III

The `RIPS FlatInstanceDescriptor` for a `Trigger Communication Graph` is only applicable, if the trigger communication configures a transformer according [SWS_Rte_08794] (inter ECU). ⌋*(SRS_Rte_00317)*

2.19 SWS_SAEJ1939DiagnosticCommunicationManager

[constr_SWS_J1939Dcm_CONSTR_6201] ⌈ J1939DcmModeCondition shall have either a J1939DcmBswModeRef or a J1939DcmSwcModeRef or a J1939DcmSwcSRDataElementRef as external reference.

⌋*()*

[constr_SWS_J1939Dcm_CONSTR_6202] ⌈ The values J1939DCM_GREATER_THAN, J1939DCM_GREATER_OR_EQUAL, J1939DCM_LESS_OR_EQUAL and J1939DCM_LESS_THAN shall not be used with a Mode reference (J1939DcmBswModeRef or J1939DcmSwcModeRef).

⌋*()*

[constr_SWS_J1939Dcm_CONSTR_6203] ⌈ The J1939DcmServiceOnlyDTCsMemoryDestinationRef shall reference an event memory assigned to the DemEventMemory Set of the current J1939DcmDemClientRef.

⌋*()*

2.20 SWS_SPIHandlerDriver

[constr_SWS_Spi_CONSTR_00001] DRATF ⌈ The ECUC partitions referenced by SpiKernelEcucPartitionRef shall be a subset of the ECUC partitions referenced by Spi EcucPartitionRef.

⌋()

[constr_SWS_Spi_CONSTR_00002] DRAFT ⌈ The ECUC partitions referenced by SpiDeviceEcucPartitionRef shall be a subset of the ECUC partitions referenced by Spi EcucPartitionRef.

⌋()

2.21 SWS_TTCANDriver

[SWS_TtCan_CONSTR_00001] ⌈ The ECUC partitions referenced by `CanTTControllerEcucPartitionRef` shall be a subset of the ECUC partitions referenced by `CanEcucPartitionRef` . ⌋()

[SWS_TtCan_CONSTR_00002] ⌈ `CanTTController` and `CanTrcvChannel` of one communication channel shall all reference the same ECUC partition. ⌋()

2.22 SWS_WatchdogManager

[constr_SWS_WdgM_CONSTR_6500] Interface provision in MCU driver ⌈ The parameter WdgMImmediateReset [ECUC_WdgM_00339] may only be set to TRUE if the McuPerformResetApi (defined in SWS_Mcu_Driver) is set to TRUE.

⌋()

[constr_SWS_WdgM_CONSTR_6501] Only non-trusted OS-Application can be restarted ⌈ The WdgM only supports the partition restart of EcuC Partitions (Wdg MEcucPartitionRef) which are linked to non-trusted OS-Applications..

⌋()

[constr_SWS_WdgM_CONSTR_6502] ⌈ A unique Supervised Entity identifier for each Supervised Entity is provided in configuration parameter WdgMSupervisedEntityID (see [ECUC_WdgM_00304]). The Identifier shall be unique in the scope of the Watchdog Manager module.

⌋()

[constr_SWS_WdgM_CONSTR_6503] ⌈ Each BSW module shall use its module ID as the Supervised Entity ID.

⌋()

[constr_SWS_WdgM_CONSTR_6504] ⌈ No SW-Cs shall have as Supervised Entity ID a value of any BSW Module ID, regardless which BSW Modules are deployed.

⌋()

[constr_SWS_WdgM_CONSTR_6505] ⌈ Deadline Supervision (WdgMDeadlineSupervision) of a Supervised Entity shall refer to Checkpoints (WdgMDeadlineStartRef, WdgMDeadlineEndRef) that both belong to that Supervised Entity. In other words, any of the referred Checkpoints shall not belong to other Supervised Entities.

⌋()

[constr_SWS_WdgM_CONSTR_6506] ⌈ Internal Transitions (see WdgMInternalTransition) in a Supervised Entity shall not connect Checkpoints that do not both belong to the same Supervised Entity.

⌋()

[constr_SWS_WdgM_CONSTR_6507] ⌈ A Checkpoint shall not belong to more than one Internal Graph.

⌋()

[constr_SWS_WdgM_CONSTR_6508] ⌈ A Checkpoint shall not belong to an External Graph and to an Internal Graph; this applies across all modes.

⌋()

[constr_SWS_WdgM_CONSTR_6509] ⌈ In a given mode, a Checkpoint shall not belong to more than one External Graph.

⌋()

[constr_SWS_WdgM_CONSTR_6510] ⌈ The following shall be available for the operation supervision functions of Watchdog Manager:

- availability of initialized Wdg Interface,
- availability of initialized OS,
- initialized WdgM - by invocation of WdgM_Init() function.

⌋()

[constr_SWS_WdgM_CONSTR_6511] ⌈ It shall be ensured by the callers of Wdg M module, that the functions WdgM_DeInit, WdgM_Init and WdgM_SetMode are not invoked concurrently to WdgM_MainFunction.

⌋()

[constr_SWS_WdgM_CONSTR_6512] ⌈ Any ordered set of two Checkpoints shall not have more than one Deadline Supervision (WdgMDeadlineSupervision) defined.

⌋()

2.23 SWS_WirelessEthernetDriver

[constr_SWS_WEth_CONSTR_00241] DRAFT ⌈ The module will operate as an independent instance in each of the partitions, means the called API will only target the partition it is called in.

⌋()

2.24 SWS_WirelessEthernetTransceiverDriver

[constr_SWS_WEthTrcv_CONSTR_00095] DRAFT ⌈ The module will operate as an independent instance in each of the partitions, means the called API will only target the partition it is called in.

⌋()

[constr_SWS_WEthTrcv_CONSTR_00096] DRAFT ⌈ WEthTrcvConfig and WEthCtrl Config of one communication channel shall all reference the same ECUC partition.

⌋()

[constr_SWS_WEthTrcv_CONSTR_00097] DRAFT ⌈ The ECUC partitions referenced by WEthTrcvConfigEcucPartitionRef shall be a subset of the ECUC partitions referenced by WEthTrcvEcucPartitionRef.

⌋()

2.25 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 must 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 must be identical to BswModuleEntity . implementedEntry . executionContext
- BswModuleEntity . callPoint . calledEntry . callType must have the value ’regular’ or ’callback’

⌋()

[constr_4016] BswCalledEntity constraints ⌈

- BswCalledEntity . implementedEntry . callType must 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 must be ’scheduled’
- BswModuleEntity . implementedEntry . executionContext must be ’task’

⌋()

[constr_4018] BswInterruptEntity constraints ⌈

- BswInterruptEntity . implementedEntry . callType must be ’interrupt’
- BswInterruptEntity . implementedEntry . executionContext must be ’interruptCat1’ if and only if BswInterruptEntity . interruptCategory is ’Cat1’

- BswInterruptEntity . implementedEntry . executionContext must 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 [2] , if applicable [2] . Otherwise (e.g. for ICC2 clusters) the identifier must either be empty or chosen differently from the ones given in [2] .

⌋()

[constr_4020] Categories of BswModuleDescription ⌈ Only categories listed in table 2.2 are allowed. Other values or an empty value are not allowed.

⌋()

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).

Table 2.2: BSWMD Categories

[constr_4021] Implementation policy of function pointer target ⌈
A BswModuleEntry can only be used as target of a function pointer (SwPointerTargetProps . functionPointerSignature), if its swServiceImplPolicy is 'standard' .

⌋()

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

- BswModuleEntity . implementedEntry must refer to an element declared as implementedEntry of the enclosing BswModuleDescription
- BswModuleEntity . callPoint . calledEntry - where callPoint is instantiated from BswDirectCallPoint - must refer to an element declared as expectedEntry or implementedEntry of the enclosing BswModuleDescription .
- BswModuleEntity . callPoint . calledEntry - where callPoint is instantiated from BswSynchronousServerCallPoint or BswAsynchronousServerCallPoint - must refer to an element declared as requiredClientServerEntry of the enclosing BswModuleDescription .
- BswModuleEntity . callPoint - where callPoint is instantiated from BswAsynchronousServerCallResultPoint - must refer to an BswAsyn-

[2]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.

chronousServerCallPoint declared in turn as callPoint of the same BswModuleEntity .

- BswModuleEntity . issuedTrigger must refer to an element declared as releasedTrigger of the enclosing BswModuleDescription
- BswModuleEntity . managedModeGroup must refer to an element declared as providedModeGroup of the enclosing BswModuleDescription
- BswModuleEntity . accessedModeGroup must refer to an element declared as requiredModeGroup of the enclosing BswModuleDescription
- BswModuleEntity . dataSendPoint . accessedVariable must refer to an element declared as providedData of the enclosing BswModuleDescription
- BswModuleEntity . dataReceivePoint . accessedVariable must refer to an element declared as requiredData of the enclosing BswModuleDescription
- an accessedModeGroup should be allowed to refer to an element declared as providedModeGroup

⌋()

[constr_4023] External trigger must belong to the interface ⌈ A BswExternalTriggerOccurredEvent must refer to a Trigger that is declared via BswModuleDescription . 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 must 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 must be referred as BswModuleDescription . providedModeGroup by the same module.

⌋()

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

⌋()

[constr_4029] Measured stack usage ⌈ The attribute values of MeasuredStackUsage must fulfill:
minimumMemoryConsumption <= averageMemoryConsumption <= maximumMemoryConsumption

⌋()

[constr_4030] Measured heap usage ⌈ The attribute values of MeasuredHeapUsage must fulfill:
minimumMemoryConsumption <= averageMemoryConsumption <= maximumMemoryConsumption

⌋()

[constr_4031] Analyzed execution time ⌈ The attribute values of AnalyzedExecutionTime must fulfill:
bestCaseExecutionTime <= bestCaseExecutionTime

⌋()

[constr_4032] Measured execution time ⌈ The attribute values of MeasuredExecutionTime must fulfill:
minimumExecutionTime <= nominalExecutionTime <= maximumExecutionTime

⌋()

[constr_4033] Simulated execution time ⌈ The attribute values of SimulatedExecutionTime must fulfill:
minimumExecutionTime <= nominalExecutionTime <= maximumExecutionTime

⌋()

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

⌋()

[constr_4038] bswModuleDependency must refer to a different module ⌈

- BswModuleDescription . bswModuleDependency . targetModuleId (if given) must 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) must 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 must have same type ⌈ SwcBswSynchronizedModeGroupPrototype can only refer to equally typed ModeDeclarationGroupPrototype s, i.e. which have identical ModeDeclarationGroup s.

⌋()

[constr_4041] Synchronized mode groups must 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 must 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 must exist.
- If category is SINGLE_POINTERED , a referenceTable must exist.
- If category is INITIALIZED_RAM , one or more elementGroup s must 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` must 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` must be such that they result in a pointer declaration.

⌋()

[constr_4054] Unambiguous links to addressing method ⌈ `MemorySection . executableEntity` must not be defined, if `MemorySection . swAddrMethod` represents a data section. `MemorySection . executableEntity` must 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 must 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 must 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` must also specify an `instanceIn-`

Memory . The target of the latter (i.e. upper level) instanceInMemory must 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 must 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 must 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 . flatInstanceDescriptor '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 . locMeasurementSet 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 must be consistent ⌈ In the case that a RunnableEntity is mapped to a BswSchedulableEn-

tity the RTE Generator emits an Entry Point Prototype only for 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 must be identical to the symbol of bswEntity as defined in [TPS_BSWMDT_04138] .
- swcRunnable . minimumStartInterval must be identical to bswEntity . minimumStartInterval .
- swcRunnable . canBeInvokedConcurrently must be identical to bswEntity . implementedEntry . isReentrant .
- swcRunnable . swAddrMethod must either be empty or must 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 must have identical shortName if they define the same bitPosition and must 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 must belong to the same BswImplementation .
- The DependencyOnArtifact referred by this link must be aggregated by BswImplementation in the role requiredArtifact .
- The DependencyOnArtifact referred by this link must 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 . role McDataAccessDetails or as the base of any McDataAccessDetails . role McDataAccessDetails 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 reentrancy values are identical. These values are taken from the attribute `isReentrant` or, if this is undefined, from `encapsulatedEntry` . `isReentrant` .
- 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 completely identical attributes.

⌋()

[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 [3]) 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` . `calibrationAccess` shall be set to `notAccessable` .

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` must have attribute values as follows:

- `callType` must be `regular` or `callback` .
- `executionContext` must 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 retrictions 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 atttribute 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 must have the attribute isSynchronous = true .
- The BswModuleClientServerEntry aggregated as calledEntry in a BswAsynchronousServerCallPoint must 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 must by aggregated by this BswInternalBehavior .

⌋()

[constr_4081] Mode group used by BSW mode manager error event ⌈ The ModeDeclarationGroupPrototype used by BswModeManagerErrorEvent must 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 BswDistinguishedPartition 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 identi-

cal 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 ⌈ For example, if we take the fourth column in table 2.3 , the invocation of an ExecutableEntity with an interruptCat1 BswExecutionContext 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 OnEntry 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 [4]

⌋()

0.9 caller's BswExecutionContext [3]	callee's BswExecutionContext [4]				
	task	**interruptCat2**	**interruptCat1**	**hook**	**unspecified**
task	Supported	Supported	Supported		Supported
interruptCat2		Supported	Supported		Supported
interruptCat1			Supported		Supported
hook					
unspecified	Supported				Supported

Table 2.3: Possible invocation of ExecutableEntity s by direct function call dependent from BswExecutionContext

[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.

⌋()

[3] The execution context of a RunnableEntity is considered as task
[4] The execution context of a RunnableEntity is considered as task

[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 BswInternalBehavior 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` . `flatInstanceDescriptor` ⌈

- 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 readOnly .

⌋()

[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 ⌈ In case a BSW module or Software Component is split into allocatable memory parts the SectionNamePrefix shall be set in the following form correspondingly:

- For BSW moduel: <MIP>_<feature>
- For Software Component: <software-component symbol name>_<feature>

where:

- <MIP>: is the capitalized Module Implementation Prefix
- <software-component symbol name>: is the symbol of the software component according to [TPS_SWCT_01000] in [3]
- <feature>: is the name of the sub-feature in the module or SWC 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.

⌋()

2.26 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 .

⌋()

[constr_1325] Allowed attributes of SwDataDefProps for DiagnosticDataElement . swDataDefProps ⌈ The allowed attributes of SwDataDefProps for the aggregation in the role DiagnosticDataElement . swDataDefProps are defined in table 2.4 .

⌋()

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

▽

△

Attributes of SwDataDefProps	DiagnosticDataElement . swDataDefProps
swTextProps	N/A
swValueBlockSize	N/A
unit	D
valueAxisDataType	N/A

Table 2.4: Allowed attributes of SwDataDefProps for DiagnosticDataElement . swDataDefProps

[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 .

⌋()

[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 .

⌋()

[constr_1328] Consistency of DiagnosticEcuInstanceProps . ecuInstance and DiagnosticServiceTable . ecuInstance ⌈ Each DiagnosticServiceTable 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 DiagnosticEcuInstanceProps . ecuInstance by a DiagnosticEcuInstanceProps referenced by the mentioned DiagnosticContributionSet **if** the respective DiagnosticContributionSet is of category DIAGNOSTICS_ECU_EXTRACT .

⌋()

[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 DiagnosticContributionSet :

- The subclass of DiagnosticServiceClass shall only appear once in the context 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.

⌋()

[constr_1330] Custom service identifier shall not overlap with standardized service 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 [1] .

⌋()

[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] .

⌋()

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

⌋()

[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 .

⌋()

[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] .

⌋()

[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).

⌋()

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

⌋()

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

⌋()

[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.

⌋()

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

⌋()

[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 .

⌋()

[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 .

⌋()

[constr_1342] Possible values for DiagnosticSecurityAccess . requestSeedId ⌈ The value of the attribute DiagnosticSecurityAccess . requestSeedId **shall only be set to an odd number** [5] .

The supported value range consists of the following list:

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

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

- **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**

⌋()

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

⌋()

[constr_1344] Condition for the identification of data types of attributes DiagnosticServiceDataMapping . mappedDataElement and DiagnosticServiceDataMapping . diagnosticDataElement ⌈ Both DiagnosticServiceDataMapping . mappedDataElement and DiagnosticServiceDataMapping . diagnosticDataElement shall be typed by either of the following options:

- ApplicationPrimitiveDataType where the value of attribute category is set to VALUE .
- ImplementationDataType where the value of attribute category is set to VALUE or to TYPE_REFERENCE that eventually resolves to an ImplementationDataType where attribute category is set to VALUE .

⌋()

[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 .

⌋()

[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

⌋()

[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

⌋()

[constr_1349] Value of udsDtcValue shall be unique ⌈ The value of udsDtcValue shall be unique to any other DTC and DTC group value.

⌋()

[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.

⌋()

[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 [1] .

⌋()

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

⌋()

[constr_1353] Applicability of [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 .

⌋()

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

⌋()

[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 [1] .

⌋()

[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 [1] .

⌋()

[constr_1359] Existence of attribute `DiagnosticDebounceAlgorithmProps.debounceCounterStorage` ⌈ The attribute `DiagnosticDebounceAlgorithmProps.debounceCounterStorage` shall only exist if the aggregation `DiagnosticDebounceAlgorithmProps.debounceAlgorithm` actually aggregates a `DiagEventDebounceCounterBased`

⌋()

[constr_1360] Usage of `DiagEventDebounceMonitorInternal` is not supported in the context of `DiagnosticDebounceAlgorithmProps` ⌈ The usage of the meta-class `DiagEventDebounceMonitorInternal` for the aggregation in the role `DiagnosticDebounceAlgorithmProps.debounceAlgorithm` is not permitted.

⌋()

[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 the same `DiagnosticEvent` are defined, then the `Enable Condition Group` mapping shall be regarded as defective.

⌋()

[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 the same `DiagnosticEvent` are defined, then the `Storage Condition Group` mapping shall be regarded as defective.

⌋()

[constr_1365] Multiplicity of `DiagnosticResponseOnEvent.event` ⌈ The multiplicity of `DiagnosticResponseOnEvent.event` shall not exceed the upper bound **255** .

⌋()

[constr_1366] Event ID in the context of diagnostic service `ResponseOnEvent` shall be unique ⌈ The value of `DiagnosticResponseOnEvent.event.dataIdentifier.id` shall be unique within the context of a given `DiagnosticResponseOnEvent` .

⌋()

[constr_1376] Multiplicity of reference `DiagnosticTroubleCodeProps.memoryDestination` ⌈ For every given `DiagnosticTroubleCodeProps` , the reference in the role `DiagnosticTroubleCodeProps.memoryDestination` **shall not exceed** the upper multiplicity 2. [constr_1377] applies.

⌋()

[constr_1377] Existence of reference `DiagnosticTroubleCodeProps.memoryDestination` ⌈ The reference `DiagnosticTroubleCodeProps.memoryDestination` shall **only** have the upper multiplicity 2 **if one (and only one) of** the referenced `DiagnosticTroubleCodeProps.memoryDestination` is a `DiagnosticMemoryDestinationMirror` .

⌋()

[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`

⌋()

[constr_1379] Existence of `DiagnosticMemoryDestinationPrimary` ⌈ Within the scope of one `DiagnosticContributionSet` only one `DiagnosticMemoryDestinationPrimary` shall exist.

⌋()

[constr_1380] Existence of `DiagnosticMemoryDestinationMirror` ⌈ Within the scope of one `DiagnosticContributionSet` only one `DiagnosticMemoryDestinationMirror` shall exist.

⌋()

[constr_1394] Value of `DiagnosticDataElement.maxNumberOfElements` depending on its existence ⌈ If the attribute `DiagnosticDataElement.maxNumberOfElements` exists then its value shall be greater than 0.

⌋()

[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 .

⌋()

[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 .

⌋()

[constr_1411] Existence of DiagnosticMemoryIdentifier . memoryHighAddressLabel vs. DiagnosticMemoryIdentifier . memoryHighAddress ⌈ At most **one** of the attributes in the following list shall exist:

- 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:

- DiagnosticMemoryIdentifier . memoryLowAddressLabel
- DiagnosticMemoryIdentifier . memoryLowAddress

⌋()

[constr_1419] Value of DiagnosticSecurityLevel . accessDataRecordSize ⌈ If the attribute DiagnosticSecurityLevel . accessDataRecordSize exists then its value shall be greater than zero.

⌋()

[constr_1421] Consistency of DiagnosticDynamicallyDefineDataIdentifierClass . subfunction ⌈ The values of DiagnosticDynamicallyDefineDataIdentifierClass . subfunction shall not repeat, i.e. every value of Diag-

nosticDynamicallyDefineDataIdentifierSubfunctionEnum shall at most appear once in the subfunction attribute.

⌋()

[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 DiagnosticEventPortMapping . swcServiceDependencyInSystem / swcFlatServiceDependency . serviceNeeds is a DiagnosticEventNeeds .

⌋()

[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 .

⌋()

[constr_1448] Interval of DiagnosticParameterIdentifier . id ⌈ The allowed interval for values of DiagnosticParameterIdentifier . id shall not exceed [0..255].

⌋()

[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 .

⌋()

[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 .

⌋()

[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` .

⌋()

[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` .

⌋()

[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`

⌋()

[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` .

⌋()

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

⌋()

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

⌋()

[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 . memoryDestination .

⌋()

[constr_1458] Reference to DiagnosticMemoryDestination ⌈ A DiagnosticMemoryDestination that is referenced by a DiagnosticTroubleCodeJ1939 . dtcProps . memoryDestination 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 .

⌋()

[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
- freezeFrameContent
- memoryDestination
- extendedDataRecord
- aging

⌋()

[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 .

⌋()

[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 .

⌋()

[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 .

⌋()

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

⌋()

[constr_1465] Allowed values of compareType in the context of a DiagnosticEnvDataCondition ⌈ Within the context of a DiagnosticEnvDataCondition **all values** of DiagnosticCompareTypeEnum are supported for the inherited attribute compareType .

⌋()

[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

⌋()

[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.

⌋()

[constr_1470] Value of DiagnosticParameter . bitOffset ⌈ The value of DiagnosticParameter . bitOffset shall only be set to a multiple of 8.

⌋()

[constr_1472] Existence of DiagnosticDataIdentifier . supportInfoByte ⌈ The attribute DiagnosticDataIdentifier . supportInfoByte **shall not exist** if the value of DiagnosticDataIdentifier . id is **outside the range 0xF400-0x F4FF** .

⌋()

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

⌋()

[constr_1510] extendedDataRecord . recordNumber shall be unique within mirror fault memory ⌈ For all DiagnosticTroubleCodeProps that refer to Diag-

nosticMemoryDestinationMirror in the role memoryDestination there shall be no two extendedDataRecord . recordNumber with the same value.

⌋()

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

⌋()

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

⌋()

[constr_1513] freezeFrame . recordNumber shall be unique within mirror fault memory ⌈ For all DiagnosticTroubleCodeProps that refer to DiagnosticMemoryDestinationMirror in the role memoryDestination there shall be no two freezeFrame . recordNumber with the same value.

⌋()

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

⌋()

[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).

⌋()

[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 PortPro-

totype that in turn is referenced by a RoleBasedPortAssignment where attribute role is set to the value IOControlRequest .

⌋()

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

⌋()

[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.

⌋()

[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.

⌋()

[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 .

⌋()

[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 .

⌋()

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

⌋()

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

⌋()

[constr_1633] Existence of `DiagnosticResponseOnEvent . event` vs. `DiagnosticResponseOnEvent . responseOnEventAction` ⌈ The existence of attributes `DiagnosticResponseOnEvent . event` vs. `DiagnosticResponseOnEvent . responseOnEventAction` is mutually exclusive, i.e. one shall only exist if the other does not exist.

⌋()

2.27 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 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 . relatedTraceItem 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_5015] Multiplicity of `multiplicityConfigClass` ⌈ The multiplicity of the attribute `EcucCommonAttributes . multiplicityConfigClass` shall not exceed 3.

⌋()

[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` 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.

⌋()

2.28 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.

⌋()

2.29 TPS_FeatureModelExchangeFormat

[constr_5001] `FMFeatureRelation` shall not establish self-references ⌈ A `FMFeatureRelation` that is aggregated by a `FMFeature` f shall not reference f in the role `feature` . In other words: self-references are not allowed.

⌋()

[constr_5002] `FMFeatureSelectionSet` shall not have cycles in the `include` relation ⌈ Let S be a `FMFeatureSelectionSet` and let G be the *inclusion graph* for all `FMFeatureSelectionSet` s as defined in [TPS_FMDT_00032] . There shall be no cycles in the inclusion graph.

⌋()

[constr_5003] `FMFeatureSelectionSet` shall not overwrite the state of included features ⌈ Let S be a `FMFeatureSelectionSet` that aggregates a `FMFeatureSelection` that has the `state` s and which refers to a `FMFeature` f in the role `feature` . Furthermore, let S_1 be a `FMFeatureSelectionSet` that aggregates a `FMFeatureSelection` that has the `state` s_1 and refers *to the same `FMFeature`* f in the role `feature` . Finally assume that S refers to S_1 in the role `include` .

Then the following conditions shall hold:

1. If the value of the attribute `state` of s_1 is `undecided` , then the value of the attribute `state` of s may be one of `selected` , `deselected` , and `undecided` .

2. If the value of the attribute `state` of s_1 is `selected` or `deselected` , then the value of the attribute `state` of s shall be the same as the attribute `state` in s_1 , or `undecided` .
3. Any other constellation is considered an error.

⌋()

[constr_5005] `FMFeature` shall not be referenced from more than one `FMFeatureDecomposition` ⌈ Let f be a `FMFeature` that is referenced from a `FMFeatureDecomposition` in the role `feature` . Then no other `FMFeatureDecomposition` shall reference f in the role `feature` .

⌋()

[constr_5007] `FMFeature` shall only be referenced from one `FMFeatureModel` in the role `feature` ⌈ Let f be a `FMFeature` , and F , F' be `FMFeatureModel` s where F references f in the role `feature` , and F' also references f in the role `feature` . Then $F = F'$.

⌋()

[constr_5008] If present, the root feature shall be part of the feature model ⌈ Let r be the `FMFeature` referenced from `FMFeatureModel` in the role `root` , and $\{f_1, f_2, \ldots, f_n\}$ the set of features referenced from the same `FMFeatureModel` in the role `feature` .

Then the following condition shall hold: $r \in \{f_1, f_2, \ldots, f_n\}$.

⌋()

[constr_5009] Root feature shall be present if and only if the feature model is not empty ⌈ If a `FMFeatureModel` refers to one or more `FMFeature` elements in the role `feature` , then exactly one of them shall be referenced by `FMFeatureModel` in the role `root` .

On the contrary, if `FMFeatureModel` does not refer to any `FMFeature` s in the role `feature` , then `root` shall be empty.

⌋()

[constr_5010] `FMFeatureDecomposition` may refer to a root feature of another feature model, but only once. ⌈ Let f_A be a `FMFeature` that is referenced by `FMFeatureModel` A in the role `feature` , but is also referenced from a `FMFeatureDecomposition` that is aggregated by a `FMFeature` f_B in the role `decomposition` .

Furthermore, let B be the `FMFeatureModel` that references f_B in the role `feature` with $A \neq B$. That is, f_A and f_B belong to different feature models.

Then *both* the following conditions shall hold:

1. f_A is referenced from A in the role `root` .

2. There is no other FMFeatureDecomposition (neither in B nor in any other FMFeatureModel) that references f_B in the role feature .

⌋()

[constr_5011] FMFormulaByFeaturesAndAttributes can refer to FMFeature s and FMAttributeDef s, but not to system constants ⌈ A formula of class FMFormulaByFeaturesAndAttributes is an expression that can use FMFeature s and FMAttributeDef s, but is not allowed to use SwSystemconst s.

⌋()

[constr_5013] Attributes min and max of FMFeatureDecomposition reserved for category MULTIPLEFEATURE ⌈ The optional attributes min and max of FMFeatureDecomposition are only allowed to be present if the category of the FMFeatureDecomposition is MULTIPLEFEATURE .

⌋()

[constr_5018] FMFeatureSelectionSet shall not include the same feature twice ⌈ Let $\{s_1, s_2, \ldots, s_n\}$ be the set of FMFeatureSelection elements that are aggregated by a FMFeatureSelectionSet in the role selection . Furthermore, for each s_i , let f_i be the FMFeature that is referred to in the role feature . Then the following condition shall hold true:

$$\forall i, j \in \{1, 2, \ldots, n\} : i \neq j \Rightarrow f_i \neq f_j$$

⌋()

[constr_5019] FMFeatureModel shall not contain the same FMFeature twice ⌈ Let F be a FMFeatureModel , and let f, f' be FMFeature s that are referenced from F in the role feature . Then $f \neq f'$.

⌋()

[constr_5020] Every FMFeature shall be contained in a FMFeatureModel ⌈ For every FMFeature f , there shall be a FMFeatureModel that refers to f in the role feature .

⌋()

[constr_5021] The underlying graph of a feature model shall be a tree. ⌈ Let F be a FMFeatureModel and G be the underlying graph of F as defined in [TPS_FMDT_00034] . Then G shall be a tree. Hence, we also refer to G as the *underlying tree* of F .

⌋()

[constr_5022] The root feature of a FMFeatureModel refers to the root of the underlying tree. ⌈ Let F be a FMFeatureModel and G be the underlying tree of F as defined in [TPS_FMDT_00034] . Furthermore, let r be the FMFeature referred to by the root feature of the FMFeatureModel .

Then the node in G which corresponds to r is the root of the tree G .

⌋()

[constr_5023] **FMFeatureSelectionSet may only refer to FMFeature s from the associated FMFeatureModel** ⌈ Let S be a FMFeatureSelectionSet , and $\{f_1, f_2, \ldots, f_n\}$ be its *feature set* ([TPS_FMDT_00009]). Furthermore, let $\{g_1, g_2, \ldots, g_m\}$ be the combined *feature sets* of the FMFeatureModel s to which S refers to in the role featureModel .

Then the following condition shall hold: $\{f_1, f_2, \ldots, f_n\} \subseteq \{g_1, g_2, \ldots, g_m\}$.

⌋()

[constr_5024] **FMFeatureSelectionSet shall not include itself** ⌈ Let S be a FMFeatureSelectionSet and let S' be the FMFeatureSelectionSet to which S refers to in the role include .

Then the following condition shall hold: $S \neq S'$.

⌋()

[constr_5025] **FMFeatureSelectionSet shall not overwrite the state of included features** ⌈ Let S be a FMFeatureSelectionSet that aggregates a FMFeatureSelection that has the state s and which refers to a FMFeature f in the role feature . Furthermore, let S_1 (S_2) be a FMFeatureSelectionSet that aggregates a FMFeatureSelection that has the state s_1 (s_2) and refers *to the same FMFeature* f in the role feature . Finally assume that S refers to S_1 *and* S_2 in the role include .

Then the following conditions shall hold:

1. If the values of the attributes state of s_1 and s_2 are both undecided , then the value of the attribute state of s may be selected , deselected or undecided .
2. If the value of the attribute state of s_1 is undecided and the value of the attribute state of s_2 is selected or deselected , then the value of the attribute state of s shall be the same as the attribute state in s_2 , or undecided .
3. If the value of the attribute state of s_2 is undecided and the value of the attribute state of s_1 is selected or deselected , then the value of the attribute state of s shall be the same as the attribute state in s_1 , or undecided .
4. If the values of the attributes state of s_1 and s_2 are both either selected or deselected , then the value of the attribute state of s shall be the same as in attribute s_1 , or undecided .
5. Any other constellation is considered an error.

⌋()

[constr_5026] Semantics of attributes `max` and `min` in class `FMAttributeDef` ⌈ The following conditions shall hold for all instances of the class `FMAttributeDef` :

- `min` $\leq$ `defaultValue` $\leq$ `max` (`min` and `max` are both closed intervals)
- `min` $<$ `defaultValue` $\leq$ `max` (`min` is an open interval, `max` is a closed interval)
- `min` $<$ `defaultValue` $<$ `max` (`min` and `max` are both open intervals)
- `min` $\leq$ `defaultValue` $<$ `max` (`min` is a closed interval, `max` is an open interval)

⌋()

[constr_5027] Semantics of attributes `max` and `min` of `FMAttributeDef` in class `FMAttributeValue` ⌈ Let v be the attribute `value` of an `FMAttributeValue` V that refers to `FMAttributeDef` D in the role `definition` . Furthermore, let min and max be the values of the attributes `min` and `max` of D .

The following condition shall hold true:

$$min \leq v \leq max$$

⌋()

[constr_5028] Only one `FMAttributeValue` per `FMAttributeDef` ⌈ Let S be a `FMFeatureSelectionSet` whose `FMFeatureSelection` s aggregate `FMAttributeValue` s $\{v_1, v_2, \ldots, v_n\}$ in the role `attributeValue` . For each v_i , let f_i be the `FMFeature` to which v_i refers to in the role `attributeDef` . Then the following condition shall hold:

$$\forall i \in \{1, \ldots, n\} : i \neq j \Rightarrow f_i \neq f_j$$

⌋()

2.30 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 *pure 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_2503] Bound model shall be compliant to the pure meta model ⌈ The *completely bound M1 model* [6] shall adhere to the *pure meta model* with respect to consistency rules and semantic constraints defined in the related template specifications. Especially, the multiplicities in the bound model shall conform to the multiplicities and the constraints of the *pure meta 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_2506] Attributes in property set pattern ⌈ On M1 level, let C be the set of attributes (or aggregated elements [7]) that would have been in the original [8] `{PropertySetClass}` object, and $C_1, \ldots, C_n$ be the respective sets of attributes in the `{PropertySetClass}Conditional` objects **for a given variant** . Also, let C' be the set of non-optional attributes, e.g., those with a lower multiplicity of 1.

We define the following constraints:

$$\forall C_i, C_j \text{ in the given variant} : C_i \cap C_j = \emptyset$$
$$C' \subseteq C_1 \cup C_2 \cup \ldots C_n \subseteq C$$

⌋()

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

⌋()

[constr_2508] Name space of `shortName` ⌈ The content of `shortName` needs to be unique (case insensitive) within a given `Identifiable` . 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

[6] Completely bound includes post build!

[7] The constraints defined in this section apply to attributes as well as aggregates elements, due to the close relationship of the two in the AUTOSAR meta model. For simplicity, the rest of this section talks about “attributes” only.

[8] In this context, “original” means `{PropertySetClass}` without the stereotype `atpVariation` . In other words, “original” means “as in the pure meta model”.

{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 package.

⌋()

[constr_2509] ReferenceBase needs to be unique in a package ⌈ The shortLabel of a reference base needs to be unique in (not within) a package. Note that it is not necessary to be unique within (to say in deeper levels) of a package.

⌋()

[constr_2510] only one default ReferenceBase ⌈ Only one ReferenceBase per level can be marked as default (default ="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 {Part Class} the shortLabel is unnecessary. In case the shortName of the {Part Class} is unique in the context of the {WholeClass} the shortLabel is unnecessary.

⌋()

[constr_2515] Categories of packages shall not conflict ⌈ If a non empty category is defined for a package, then all sub packages shall have empty category or the same category. See table **??** . Additionally, the "Rules for references between elements in packages with specific categories" shall apply. See table **??** .

⌋()

[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 `PredefinedVariant` plus its `includedVariant` s references more than one `SwSystemconstantValueSet` all `value` attributes in `SwSystemconstValue` s for a particular `SwSystemconst` shall be identical.

⌋()

[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 `atpSplitable` shall be all together 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_00381] .

⌋()

[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 `DocumentationContext` .

⌋()

[constr_2534] Limits of unlimited Integer ⌈ 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 (0x800000000000000014) and the max value can be up to 9223372036854775807 (`0x7fffffffffffffff`).

If an unsigned value is represented the min value can be down to 0 and the max value can be up to 18446744073709551615 (`0xffffffffffffffff`).

⌋()

[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 partial models ⌈ Ordered collections cannot be split. In other words: In opposite to 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 `BlueprintDerivationTime` by the meta model.

⌋()

[constr_2558] If `vh.latestBindingTime` is `BlueprintDerivationTime` then there shall only be `blueprintCondition` / `blueprintValue` ⌈ `VariationPoint` s with `vh.latestBindingTime` restricted to `BlueprintDerivation` shall not have `swSysCond` nor `postbuildVariantCondition` .

⌋()

[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 `blueprintValue` is specified, then the `value` defined by the `AttributeValueVariationPoint` 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_2574] `globalInPackage` for global elements only ⌈ `ReferenceBase` . `globalInPackage` is allowed only if `isGlobal` is set to true.

⌋()

[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]):

ConditionByFormula . bindingTime ≤ *aggregation* . 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 ≤ *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 ≤ *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 atpVariation (see [TPS_GST_00190] , [TPS_GST_00220] , [TPS_GST_00221]):

VariationPoint . bindingTime ≤ *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` , `arReleaseVersion` , `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 `atpMixed` 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 `ImplementationDataTypeElement` ⌈ If the `target` references an `ImplementationDataTypeElement` 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_2605] If a SdgClass is referenced then it shall have a caption ⌈ `destSdg . caption` ==true

⌋()

[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 be an `atpFeature` of 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_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 .

⌋()

2.31 TPS_SafetyExtensions

[constr_6200] Safety goals have no decomposed ASIL ⌈ If a safety requirement is of type SAFETY_GOAL the valid values of the ASIL attribute are restricted to: QM, A, B, C, or D .

⌋()

[constr_6201] Consistency of ASIL values ⌈ The ASIL of AUTOSAR elements and allocated safety requirements should be *consistent* . An ASIL is consistent if the value at an element is the same or higher of the maximum ASIL of allocated safety requirements.

⌋()

[constr_6202] Decomposition into two safety requirements ⌈ A decomposition as specified by [TPS_SAFEX_00302] shall be specified at exactly two decomposing safety requirements (not more) for each decomposed requirement.

⌋()

[constr_6203] Decomposing only one safety requirement ⌈ Each decomposing requirement specified according to [TPS_SAFEX_00302] shall decompose maximum one other requirement.

⌋()

2.32 TPS_SoftwareComponentTemplate

[constr_1000] End-to-end protection is limited to sender/receive communication ⌈ end-to-end protection applies for sender/receiver communication only

⌋()

[constr_1001] Value of dataId shall be unique ⌈ The value of the dataId shall be unique within the scope of the System .

⌋()

[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 .

⌋()

[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).

⌋()

[constr_1006] applicable data categories ⌈ Table 2.5 defines the applicable `category` s depending on specific model elements related to data definition properties.

⌋()

Category	Applicable to ...											Use Case				Description
	ApplicationArrayDataType	ApplicationRecordDataType	ApplicationPrimitiveDataType	ApplicationRecordElement	ApplicationArrayElement	ApplicationValueSpecification	ImplementationDataType	ImplementationDataTypeElement	SwServiceArg	SwSystemconst	McDataInstance	Calibration	Measurement	Communication Port Interfaces	RTE + BSW	
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		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					x [9]	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.

▽

[9] [constr_1295] applies!

△

Category	Applicable to ...											Use Case				Description
	ApplicationArrayDataType	ApplicationRecordDataType	ApplicationPrimitiveDataType	ApplicationRecordElement	ApplicationArrayElement	ApplicationValueSpecification	ImplementationDataType	ImplementationDataTypeElement	SwServiceArg	SwSystemconst	McDataInstance	Calibration	Measurement	Communication Port Interfaces	RTE + BSW	
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	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.
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		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 CURVE s, MAP s, CUBOID s, CUBE_4 s, and CUBE_5 s.
RES_AXIS			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		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	ImplementationDataType	ImplementationDataTypeElement	SwServiceArg	SwSystemconst	McDataInstance	Calibration	Measurement	Communication Port Interfaces	RTE + BSW	
MAP			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		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		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		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.
MACRO									x						x	This represents an argument to a C macro.

Table 2.5: Usage of category for Data Types

[constr_1007] Allowed attributes of SwDataDefProps for ApplicationDataType s ⌈ The allowed attributes of SwDataDefProps for Application-

DataType s and their allowed multiplicities are listed as an overview in table 2.6 .

⌋()

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
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 [10]	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 [11]			0..1		1	1	1	1	1	1	1
swRefreshTiming	x			0..1	0..1			0..1	0..1							

▽

[10] don't care

[11] This is required by [TPS_SWCT_01179] .

△

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..*										
element : ApplicationArrayElement	x	x	x				1									
ApplicationArrayElement . arraySizeSemantics	x						0..1									
ApplicationArrayElement . maxNumberOfElements	x						1									

Table 2.6: Allowed Attributes vs. `category` for `ApplicationDataType` s

[constr_1008] Applicability of `category` s `STRUCTURE` and `ARRAY` ⌈ The categories `STRUCTURE` and `ARRAY` correspond to `ApplicationCompositeDataType` s whereas all other `category` s can be applied only for `ApplicationPrimitiveDataType` s.

⌋()

[constr_1009] `SwDataDefProps` applicable to `ImplementationDataType` s ⌈ A complete list of the `SwDataDefProps` and other attributes and their multiplicities which are allowed for a given `category` is shown in table 2.7 .

⌋()

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 [12]			d/c			d/c

▽

[12] 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
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			
invalidValue	x	x	x		0..1			0..1	0..1 [13]		0..1 [14]
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											

▽

[13] 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.

[14] 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.

△

Attributes of SwDataDefProps	Root Element				Attribute Existence per Category						
	ImplementationDataType	ImplementationDataTypeElement	SwPointerTargetProps	SwServiceArg	VALUE	DATA_REFERENCE	FUNCTION_REFERENCE	TYPE_REFERENCE	STRUCTURE	UNION	ARRAY
valueAxisDataType											
Other Attributes											
subElement : ImplementationDataTypeElement	x	x							1..*	1..*	1
subElement . arraySizeSemantics	x	x									0..1
subElement . arraySize	x	x									1

Table 2.7: Allowed Attributes vs. `category` for `ImplementationDataType`

[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.

⌋()

[constr_1011] `category` of `SwBaseType` ⌈ For the attribute `SwBaseType . category` only the values `FIXED_LENGTH` and `VOID` are supported.

⌋()

[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.

⌋()

[constr_1014] Supported value encodings for `SwBaseType` ⌈ The supported values for attribute `BaseTypeDirectDefinition . baseTypeEncoding` are:

- `1C` : One’s complement
- `2C` : Two’s complement
- `BCD-P` : Packed Binary Coded Decimals
- `BCD-UP` : Unpacked Binary Coded Decimals
- `DSP-FRACTIONAL` : Digital Signal Processor
- `SM` : Sign Magnitude

- 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* [6]
- UCS-2 : Universal Character Set 2
- NONE : Unsigned Integer
- VOID : corresponds to a void in C. The encoding is not formally specified here.
- BOOLEAN : This represents an unsigned integer to be interpreted as boolean. The value shall be interpreted as true if the value of the unsigned integer is 1 and it shall be interpreted as false if the value of the unsigned integer is 0.

  A CompuMethod shall be referenced by the corresponding AutosarDataType that implements the common sense behind the boolean concept, i.e. define a TEXTTABLE with two CompuScale s: e.g. true –> 1, false –> 0.

⌋()

[constr_1015] Prioritization of SwDataDefProps ⌈ The prioritization and usage of attributes of meta-class SwDataDefProps shall follow the restrictions given in table 2.8 .

⌋()

Attributes of SwDataDefProps	Usage For			Place of Setting										
	RTE	A2L	Other Usage	ApplicationDataType	ImplementationDataType	DataPrototype	InstantiationDataDefProps	ParameterAccess	ComSpec	SwServiceArg	FlatInstanceDescriptor	McDataInstance	SwSystemconst	PerInstanceMemory
additionalNativeTypeQualifier	x		x	NA	D	I	NA	NA	NA	D	NA	S	NA	NA
annotation			x	D	A	A	A	A	A	D	NA	A	D	NA
baseType	x	x	x	NA	D	I	I	I	R	D	NA	S	M	NA
compuMethod	x	x	x	D	AI	I	I	NA	R	I	AI	S	D	NA
dataConstr	x	x	x	D	C	R	R	I	NA	R	NA	S	D	NA
displayFormat		x		D	A	R	R	I	NA	R	NA	S	D	NA
displayPresentation	x	x	x	D	A	R	R	NA	NA	NA	NA	S	NA	NA
implementationDataType	x		x	NA	D	I	I	I	NA	D	NA	NA	NA	NA
invalidValue	x	x		D	A	I	I	NA	D	NA	NA	S	NA	NA
stepSize		x		D	A	A	A	A	NA	NA	A	S	NA	NA
swAddrMethod	x	x	x	D	R	R	R	NA	NA	NA	R	NA	NA	D
swAlignment	x		x	NA	D	R	R	NA	NA	NA	NA	NA	NA	NA
swBitRepresentation		x	x	NA	NA	NA	NA	NA	NA	NA	NA	D	NA	NA
swCalibrationAccess	x	x		D	R	R	R	NA	NA	R	R	S	D	NA
swCalprmAxisSet	x	x		D	NA	I	I	I	NA	NA	NA	S	NA	NA
swCalprmAxisSet . swCalprmAxis / SwAxisGrouped . swCalprmRef		x		NA	NA	NA	D	R	NA	NA	NA	S	NA	NA
swCalprmAxisSet . swCalprmAxis / SwAxisIndividual . swVariableRef		x		NA	NA	NA	D	R	NA	NA	NA	S	NA	NA
swCalprmAxisSet . swCalprmAxis / SwAxisGrouped . sharedAxisType		x		D	NA	NA	NA	NA	NA	NA	NA	S	NA	NA
swCalprmAxisSet . swCalprmAxis / SwAxisIndividual . inputVariableType		x		D	NA	NA	NA	NA	NA	NA	NA	S	NA	NA
swCalprmAxisSet / SwAxisIndividual . unit		opt.		D	NA	I	I	I	NA	I	NA	S	NA	NA
swComparisonVariable		x		NA	NA	NA	NA	D	NA	NA	NA	S	NA	NA
swDataDependency		x	x	NA	NA	D	R	NA	NA	NA	NA	S	NA	NA
swHostVariable		x	x	NA	NA	NA	NA	NA	NA	NA	NA	D	NA	NA
swImplPolicy	x		x	D	A	A	NA	NA	NA	D	NA	NA	NA	NA
swIntendedResolution			x	D [15]	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
swInterpolationMethod			x	D	I	R	R	R	NA	NA	NA	S	NA	NA
swIsVirtual		x		NA	NA	D	R	NA	NA	NA	NA	S	NA	NA
swPointerTargetProps			x	NA	D	I	NA	NA	NA	D	NA	NA	NA	NA
swRecordLayout	x	x	x	D	NA	I	I	I	NA	NA	NA	S	NA	NA
swRefreshTiming		x		D	R	R	R	NA	NA	R	R	R	NA	NA
swTextProps		x	x	D	I	I	I	I	NA	NA	NA	S	NA	NA
swValueBlockSize		x	x	D	I	I	I	I	NA	NA	NA	S	NA	NA

▽

[15] 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 .

△

Attributes of SwDataDefProps	Usage For			Place of Setting										
	RTE	A2L	Other Usage	ApplicationDataType	ImplementationDataType	DataPrototype	InstantiationDataDefProps	ParameterAccess	ComSpec	SwServiceArg	FlatInstanceDescriptor	McDataInstance	SwSystemconst	PerInstanceMemory
swValueBlockSizeMult		x	x	D	I	I	I	I	NA	NA	NA	S	NA	NA
unit		x	x	D	I	I	I	NA	NA	I	NA	S	D	NA
valueAxisDataType		x	x	D	I	I	I	I	NA	NA	NA	S	NA	NA

Table 2.8: Usage of Attributes of SwDataDefProps

Please note that this table is (by reference) a part of [constr_1015]

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

⌋()

[constr_1017] Supported combinations of swImplPolicy and swCalibrationAccess ⌈ The table 2.9 defines the supported combinations of swImplPolicy and swCalibrationAccess attribute setting.

⌋()

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

Table 2.9: Supported combinations of swImplPolicy and swCalibrationAccess

[constr_1018] 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 .

⌋()

[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` .

⌋()

[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.

⌋()

[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` .

⌋()

[constr_1025] Avoid division by zero in rational formula ⌈ The rational formula shall not yield any division by zero.

⌋()

[constr_1026] Compatibility of `Unit` s ⌈ For data types or prototypes, units should be referenced from within the associated `CompuMethod` . But if it is referenced from within `SwDataDefProps` and/or `PhysConstrs` (for exceptional use cases) it shall be compatible (for more details please refer to [constr_1052]) to the ones referenced from the referred `CompuMethod` .

⌋()

[constr_1027] Types for record layouts ⌈ Because `ParameterDataPrototype` s have a `isOfType` -relation to `ApplicationDataType` s or `ImplementationDataType` s the related data types shall properly match to the details as specified in `swDataDefProps` .

⌋()

[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`).

⌋()

[constr_1030] **ParameterSwComponentType references ConstantSpecificationMappingSet** ⌈ ParameterSwComponentType : here the ConstantSpecificationMappingSet is directly associated by the ParameterSwComponentType .

⌋()

[constr_1031] **NvBlockSwComponentType references ConstantSpecificationMappingSet** ⌈ NvBlockSwComponentType : in this case the ConstantSpecificationMappingSet is associated with the aggregated NvBlockDescriptor .

⌋()

[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.

⌋()

[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.

⌋()

[constr_1036] Connect kinds of PortInterface s ⌈ It shall not be possible to connect PortPrototype s typed by PortInterface s of different kinds. Subclasses of DataInterface make an exception from 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.

⌋()

[constr_1038] Reference to ApplicationError ⌈ A possibleError referenced by a ClientServerOperation shall be owned by the PortInterface that also owns the ClientServerOperation .

⌋()

[constr_1039] Relevance of swImplPolicy ⌈ It is not possible to define a mapping between an element where the swImplPolicy is set to queued and an other element where the swImplPolicy is set differently.

⌋()

[constr_1040] Conversion of SenderReceiverInterface s ⌈ The conversion of elements of SenderReceiverInterface s is possible if one of the following conditions applies:

- 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.

⌋()

[constr_1043] PortInterface vs. ComSpec ⌈ The allowed combinations of a specific kind of PortInterface and a kind of ComSpec are documented in Table 2.10 .

⌋()

PortInterface	ComSpec
SenderReceiverInterface	SenderComSpec , ReceiverComSpec
ClientServerInterface	ClientComSpec , ServerComSpec
ModeSwitchInterface	ModeSwitchSenderComSpec , ModeSwitchReceiverComSpec
ParameterInterface	ParameterProvideComSpec , ParameterRequireComSpec
NvDataInterface	NvRequireComSpec , NvProvideComSpec

Table 2.10: PortInterface vs. ComSpec

[constr_1044] Applicability of DataFilter ⌈ According to the origin of DataFilter , i.e. ISO 17356-4 specification [7] , DataFilter s can only be applied to values with an integer base type.

⌋()

[constr_1045] Supported value encodings for SwBaseType in the context of PortInterface s ⌈ The supported value encodings for the usage within a PortInterface 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* [6]
- UCS-2 : Universal Character Set 2
- NONE : Unsigned Integer
- BOOLEAN : This represents an integer to be interpreted as boolean.

⌋()

[constr_1046] Applicability of [constr_1045] ⌈ [constr_1045] applies **only** if the value of the attribute isService is set to false .

⌋()

[constr_1047] Compatibility of ApplicationPrimitiveDataType s ⌈ Instances of ApplicationPrimitiveDataType are compatible if and only if one of the following conditions applies:

1. All of the following subconditions apply:
   (a) They have the same category .
   (b) The swDataDefProps 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 ApplicationRecordDataType s ⌈ Instances of ApplicationRecordDataType s are compatible 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 `SubElementMapping` exists such that a `ApplicationCompositeDataTypeSubElementRef` 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 if and only if one of the following conditions applies:

1. All of the following subconditions apply:
   (a) Their `element` s are of a compatible `AutosarDataType` s (either `ApplicationCompositeDataType` s or `ApplicationPrimitiveDataType` 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 if and only if after all type-references are resolved one of the following rules apply:

1. All of the following subconditions 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 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 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. 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 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 [16] .

4. They refer to compatible data constraints dataConstr .
5. They refer to compatible swRecordLayout s

All other attributes (e.g. swCalibrationAccess do not affect compatibility).

⌋()

[constr_1052] Compatibility of Unit s ⌈ Two Unit definitions are compatible if and only if:

1. They have compatible (see [TPS_GST_02501]) values of attributes factorSiToUnit and offsetSiToUnit .
2. They either refer to identical definitions of PhysicalDimension or neither of them associates 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
- luminousIntensityExp

are identical and **either** the shortName s are identical **or** a PhysicalDimensionMapping exists that maps one of the PhysicalDimension s in the role firstPhysicalDimension and the other PhysicalDimension in the role secondPhysicalDimension .

⌋()

[16] 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.

[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.

⌋()

[constr_1055] `ImplementationDataType` has `category VALUE` ⌈ The attributes `baseType` shall refer to a compatible `SwBaseType`

⌋()

[constr_1056] `ImplementationDataType` has `category TYPE_REFERENCE` ⌈ The `ImplementationDataType` s referenced by the attributes `SwDataDefProps` . `implementationDataType` shall be compatible .

⌋()

[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

⌋()

[constr_1058] `ImplementationDataType` has `category FUNCTION_REFERENCE` ⌈ The attributes `SwDataDefProps` . `swPointerTargetProps` . `functionPointerSignature` shall refer to `BswModuleEntry` s which each resolve to the **same function signature** .

⌋()

[constr_1059] Compatibility of data types with `category VALUE` ⌈ An `ApplicationDataType` of `category VALUE` can only be mapped/connected to an `ImplementationDataType` which also has `category VALUE` .

⌋()

[constr_1060] Compatibility of data types with `category ARRAY` , `VAL_BLK` ⌈ An `ApplicationDataType` of `category ARRAY` , `VAL_BLK` can only be mapped/connected to

- an `ImplementationDataType` of `category ARRAY` **or**
- an `ImplementationDataType` that represents a `Variable-Size Array Data Type` (see [TPS_SWCT_01610]).

⌋()

[constr_1061] Compatibility of data types with `category STRUCTURE` ⌈ An `ApplicationDataType` of `category STRUCTURE` can only be mapped/connected to an `ImplementationDataType` of `category STRUCTURE` .

⌋()

[constr_1063] Compatibility of data types with category BOOLEAN ⌈ An ApplicationDataType of category BOOLEAN can only be mapped/connected to an ImplementationDataType of category VALUE .

⌋()

[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 can only be mapped/connected to an ImplementationDataType of category STRUCTURE or ARRAY .

⌋()

[constr_1066] Forbidden mappings to ImplementationDataType ⌈ An ApplicationDataType shall never be mapped to an ImplementationDataType of category UNION , DATA_REFERENCE , or FUNCTION_REFERENCE .

⌋()

[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 if and only if one of the following conditions applies:

1. All of 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 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 PortPrototype s of different DataInterface s in the context of DelegationSwConnector s ⌈ PortPrototype s of different DataInterface s are compatible 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 PortPrototype a compatible VariableDataPrototype or ParameterDataPrototype exists in the DataInterface of the required outer PortPrototype .

The shortName of VariableDataPrototype s and ParameterDataPrototype s 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 VariableDataPrototype s or ParameterDataPrototype s 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 ParameterDataPrototype defined in the context of the SenderReceiverInterface , NvDataInterface or ParameterInterface of the provided inner PortPrototype a compatible VariableDataPrototype or ParameterDataPrototype exists in the SenderReceiverInterface , NvDataInterface or ParameterInterface of the provided outer PortPrototype .

The shortName s of VariableDataPrototype s and ParameterDataPrototype s 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 ParameterDataPrototype 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 VariableDataPrototype ⌈ Combinations of ParameterDataPrototype and VariableDataPrototype used in PortPrototype s typed by various kinds of PortInterface s shall only be allowed where Table 2.11 contains the value “yes”.

⌋()

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

Table 2.11: Overview of compatibility of ParameterDataPrototype and VariableDataPrototype

[constr_1072] Compatibility of `ModeSwitchInterface` s in the context of an `AssemblySwConnector` ⌈ `PortPrototype` s of different `ModeSwitchInterface` s are compatible 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 `ModeSwitchInterface` 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 `secondModeGroup` .
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 `ModeSwitchInterface` s are compatible 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 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 if and only if one of the following conditions applies:

1. All of 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 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 if and only if one of the following conditions applies:

1. All of 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 compatible 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 the same shortName (again allows for mapping in PortInterface s).
4. 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 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 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 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 if and only if all of 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 of 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 if they have an identical `shortName` .

⌋()

[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 `ParameterInterface` 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.

⌋()

[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 .

⌋()

[constr_1087] AssemblySwConnector inside CompositionSwComponentType ⌈ An AssemblySwConnector can only connect PortPrototype s of SwComponentPrototype s that are owned by the same CompositionSwComponentType

⌋()

[constr_1088] DelegationSwConnector inside CompositionSwComponentType ⌈ A DelegationSwConnector can only connect a PortPrototype of a SwComponentPrototype that is owned by the same CompositionSwComponentType that also owns the connected delegation PortPrototype .

⌋()

[constr_1090] WaitPoint and RunnableEntity ⌈ A single RunnableEntity can actually wait only at a single WaitPoint provided that the RunnableEntity can only be scheduled a single time [17] .

⌋()

[constr_1091] RTEEvent s that can unblock a WaitPoint ⌈ The only RTEEvent s that are qualified for unblocking a WaitPoint are:

- DataReceivedEvent
- DataSendCompletedEvent
- ModeSwitchedAckEvent
- AsynchronousServerCallReturnsEvent

⌋()

[constr_1092] ParameterSwComponentType ⌈ A ParameterSwComponentType shall never aggregate a SwcInternalBehavior and also owns exclusively PPortPrototype s of type ParameterInterface .

⌋()

[constr_1093] Definition of textual strings ⌈ An ApplicationPrimitiveDataType of category STRING shall have a swTextProps which determines the arraySizeSemantics and swMaxTextSize .

⌋()

[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 .

⌋()

[constr_1096] SwcModeSwitchEvent and WaitPoint ⌈ A RunnableEntity that has a WaitPoint shall not be referenced by a SwcModeSwitchEvent .

⌋()

[constr_1097] RunnableEntity that has a WaitPoint ⌈ A RunnableEntity that has a WaitPoint shall not be referenced by a RTEEvent that has a reference in the role disabledMode .

⌋()

[constr_1098] Mode switch and mode disabling ⌈ A SwcModeSwitchEvent shall not simultaneously reference to the same ModeDeclaration in both the roles mode and disabledMode .

[17] This constraint is valid at least in the ISO 17356-3 [8] standard where an extended task (that can have wait points) can only exist a single time in the context of the scheduler.

⌋()

[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 .

⌋()

[constr_1101] Mode-related communication ⌈ An RPortPrototype typed by ModeSwitchInterface shall not be referenced by more than one SwConnector .

⌋()

[constr_1102] ApplicationError in the scope of one SwComponentType ⌈ If a SwComponentType has PortPrototype s typed by different ClientServerInterface s with equal shortName and ApplicationError s defined then the following condition applies: ApplicationError s with the same shortName shall have **identical values** of errorCode s.

⌋()

[constr_1103] NonqueuedReceiverComSpec and enableUpdate ⌈ A NonqueuedReceiverComSpec that has attribute enableUpdate set to true may not reference a dataElement that in turn is referenced by a VariableAccess in the role dataReadAccess .

⌋()

[constr_1104] Trigger sink and trigger source ⌈ An RPortPrototype typed by a TriggerInterface shall not be referenced by more than one SwConnector s that are in turn referencing PPortPrototype s typed by TriggerInterface s that contain Trigger s with the same shortName .

⌋()

[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 .

⌋()

[constr_1106] Structure shall have at least one element ⌈ An ImplementationDataType or ImplementationDataTypeElement of category STRUCTURE shall own at least one ImplementationDataTypeElement .

⌋()

[constr_1107] Union shall have at least one element ⌈ An ImplementationDataType or ImplementationDataTypeElement of category UNION shall own at least one ImplementationDataTypeElement .

⌋()

[constr_1108] Value of `ApplicationError.errorCode` ⌈ The value of `ApplicationError.errorCode` shall not exceed the closed interval 1 .. 63. The following exception applies: **only** in case `possibleError` is supposed to represent `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.

⌋()

[constr_1110] Value of `category` in `EndToEndDescription` ⌈ The attribute `category` of `EndToEndDescription` can have the following values:

- NONE
- PROFILE_01
- PROFILE_02

⌋()

[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].

⌋()

[constr_1112] Constraints of `dataIdMode` in PROFILE_01 ⌈ In PROFILE_01, the applicable range of values for `dataIdMode` is [0 .. 3].

⌋()

[constr_1113] Existence of attributes in PROFILE_01 ⌈ In PROFILE_01, the following attributes shall exist:

- `dataLength`
- `dataId`

⌋()

[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.

⌋()

[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.

⌋()

[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.

⌋()

[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].

⌋()

[constr_1118] Existence of attributes in PROFILE_02 ⌈ In PROFILE_02, only the following attributes shall exist:

- `dataLength`
- `dataId`

⌋()

[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.

⌋()

[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].

⌋()

[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].

⌋()

[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* .

⌋()

[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.

⌋()

[constr_1129] swImplPolicy and NonqueuedReceiverComSpec ⌈ The attribute swImplPolicy of a dataElement referenced by a NonqueuedReceiverComSpec **shall not** be set to the value queued .

⌋()

[constr_1130] swImplPolicy and QueuedReceiverComSpec ⌈ The attribute swImplPolicy of a dataElement referenced by a QueuedReceiverComSpec **shall** be set to the value queued .

⌋()

[constr_1131] swImplPolicy and NonqueuedSenderComSpec ⌈ The attribute swImplPolicy of a dataElement referenced by a NonqueuedSenderComSpec **shall not** be set to the value queued .

⌋()

[constr_1132] swImplPolicy and QueuedSenderComSpec ⌈ The attribute swImplPolicy of a dataElement referenced by a QueuedSenderComSpec **shall** be set to the value queued .

⌋()

[constr_1134] Allowed structure of TEXTTABLE ⌈ physConstrs is not allowed. compuInternalToPhys shall exist with compuScale s consisting of upperLimit and lowerLimit .

⌋()

[constr_1135] Limit of vt in BITFIELD_TEXTTABLE ⌈ The separator is "|" and is forbidden in vt therefore.

⌋()

[constr_1137] Applicability of `ParameterInterface` ⌈ A `PPortPrototype` typed by a `ParameterInterface` can **only** be owned by a `ParameterSwComponentType` .

⌋()

[constr_1138] `assignedPort` and `DiagEventDebounceMonitorInternal` ⌈ The existence of an `assignedPort` in combination with a `DiagEventDebounceAlgorithm` shall only be respected for the concrete subclass `DiagEventDebounceMonitorInternal` .

⌋()

[constr_1139] `assignedPort` of `DiagEventDebounceMonitorInternal` shall refer to an `RPortPrototype` ⌈ Concerning the debouncing, the software-component acts as a client and thus the `assignedPort` defined with respect to a `DiagEventDebounceMonitorInternal` may only refer to an `RPortPrototype` . The standardized value of the `role` identifier of the `assignedPort` shall be `DiagFaultDetectionCounterPort` .

⌋()

[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.

⌋()

[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`

⌋()

[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`

⌋()

[constr_1143] `category` of `AutosarDataType` 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 `AutosarDataType`

⌋()

[constr_1144] `SensorActuatorSwComponentType` , `EcuAbstractionSwComponentType` , 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 `HwDescriptionEntity` are not allowed.

⌋()

[constr_1146] Applicability of a symbol for a `CompuScale` in C code ⌈ The `symbol` attribute shall only be provided for `CompuScale` s where the `category` of the enclosing `CompuMethod` is one of the following:

- `SCALE_LINEAR_AND_TEXTTABLE`
- `SCALE_RATIONAL_AND_TEXTTABLE`
- `TEXTTABLE`
- `BITFIELD_TEXTTABLE`

⌋()

[constr_1147] Standardized values for the attribute `category` of meta-class `PortGroup` ⌈ The following values of the attribute `category` of meta-class `PortGroup` are reserved by the AUTOSAR standard:

- `MODE_MANAGEMENT` : This represents the usage of the `PortGroup` for the purpose of mode management
- `PARTIAL_NETWORKING` : This represents the usage of the `PortGroup` for the purpose of partial networking

⌋()

[constr_1148] `PortInterface` s of `PortPrototype` s used to connect to `NvBlockSwComponentType` s ⌈ `PortInterface` s of `PortPrototype` s used to connect to `NvBlockSwComponentType` s as well as the `PortInterface` s used in the context of `NvBlockSwComponentType` s shall **always** set the value of the attribute `isService` to `false` .

⌋()

[constr_1149] `PortPrototype` s used for NV data management ⌈ A `PortPrototype` typed by a `ClientServerInterface` used for NV data management, i.e.

the interaction of ApplicationSwComponentType s with NvBlockSwComponentType s, shall be typed by ClientServerInterface s that are compatible to the particular ClientServerInterface s derived from MOD_GeneralBlueprints [9] . [constr_1148] applies.

⌋()

[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 .

⌋()

[constr_1151] Applicability of PortInterfaceMapping ⌈ A PortInterfaceMapping is only applicable and valid for a SwConnector if the two PortPrototype s which are referenced by the SwConnector are typed by the same two PortInterface s which are mapped by the PortInterfaceMapping .

⌋()

[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 CompuScale s ⌈ Compatibility requirements for CompuScale s shall only apply for CompuScale s where the category of the enclosing CompuMethod is one of the following:

- SCALE_LINEAR_AND_TEXTTABLE
- SCALE_RATIONAL_AND_TEXTTABLE
- TEXTTABLE
- TAB_NOINTP
- BITFIELD_TEXTTABLE
- LINEAR
- RAT_FUNC
- IDENTICAL

⌋()

[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).

⌋()

[constr_1155] Compatibility of CompuScale s for client-server communication ⌈ For client-server communication, the following rules apply:

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 , compuConst and symbol) are equal. If the scale has no compuConst , "names" of CompuScale s are not relevant for compatibility.

⌋()

[constr_1157] Applicability of constraints of CompuScale s ⌈ The constraints [constr_1154] , [constr_1155] , and [constr_1156] shall **only** apply in the absence of a TextTableMapping which shall take precedence regarding the compatibility if it exists.

⌋()

[constr_1158] Applicable category s for attribute ImplementationDataType . swDataDefProps . compuMethod ⌈ The definition of the reference ImplementationDataType . swDataDefProps . compuMethod is restricted to a CompuMethod of either category BITFIELD_TEXTTABLE or category TEXTTABLE (these might be seen as implementation specific in certain cases).

⌋()

[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 .

⌋()

[constr_1160] Size of Compound Primitive Data Type is variant ⌈ For Compound Primitive Data Type s (see [TPS_SWCT_01179]) where the size is subject to variation the size of the specified initValue s shall match the range of the involved SwSystemconst .

⌋()

[constr_1161] Applicability of the index attribute of Ref ⌈ The index attribute of Ref is limited to a given set if use cases as there are:

- McDataInstance . instanceInMemory
- AutosarVariableRef
- AutosarParameterRef
- FlatInstanceDescriptor / AnyInstanceRef

⌋()

[constr_1162] Compatibility of SwRecordLayout s ⌈ Two SwRecordLayout definitions are compatible if and only if all attributes **except**

- shortName
- desc
- introduction
- longName
- adminData
- annotation

are **identical** .

⌋()

[constr_1163] Compatibility of CompuMethod s ⌈ Two CompuMethod definitions are compatible 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 .

⌋()

[constr_1165] Applicability of RunnableEntityArgument ⌈ The existence of a RunnableEntityArgument is limited to RunnableEntity s triggered by a ClientServerOperation .

⌋()

[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 .

⌋()

[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 .

⌋()

[constr_1168] Compatibility of ImplementationDataType s used in the ModeRequestTypeMap ⌈ Both ImplementationDataType s shall fulfill [constr_1167] .

In addition to that, the possible numbers used for representing ModeDeclaration s on the side of the mode manager shall match the supported range of the ImplementationDataType used for representing ModeDeclaration s on the side of the mode user (see [constr_1075]).

⌋()

[constr_1169] Allowed values for Trigger . swImplPolicy ⌈ 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_1170] Interpretation of attribute maxDeltaCounterInit owned by EndToEndDescription ⌈ If EndToEndProtection . endToEndProtectionVariablePrototype . receiver is identical to the RPortPrototype . requiredComSpec . dataElement **and** RPortPrototype . requiredComSpec . maxDeltaCounterInit is defined **then** the value of RPortPrototype . requiredComSpec . maxDeltaCounterInit **shall be preferred** over the value of EndToEndProtection . endToEndProfile . maxDeltaCounterInit .

If the value of category of EndToEndDescription is set to PROFILE_01 **and either** the described correspondence rule concerning the referenced VariableDataPrototype is not fulfilled **or** RPortPrototype . requiredComSpec . maxDeltaCounterInit is not defined **then** EndToEndProtection . endToEndProfile . maxDeltaCounterInit **shall exist** .

⌋()

[constr_1171] Interpretation of attribute maxDeltaCounterInit of EndToEndDescription ⌈ If EndToEndProtection . endToEndProtectionVariablePrototype . receiver is identical to the RPortPrototype . requiredComSpec . dataElement **and** RPortPrototype . requiredComSpec . maxDeltaCounterInit is defined **then** the value of RPortPrototype . requiredComSpec . maxDeltaCounterInit **shall be preferred** over the value of EndToEndProtection . endToEndProfile . maxDeltaCounterInit .

If the value of category of EndToEndDescription is set to PROFILE_02 **and either** the described correspondence rule concerning the referenced VariableDataPrototype is not fulfilled **or** RPortPrototype . requiredComSpec . maxDeltaCounterInit is not defined **then** EndToEndProtection . endToEndProfile . maxDeltaCounterInit **shall exist** .

⌋()

[constr_1172] Allowed values of SwCalibrationAccessEnum for ModeDeclarationGroupPrototype ⌈ The only allowed values of swCalibrationAccess aggregated by ModeDeclarationGroupPrototype are notAccessible and readOnly .

⌋()

[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` .

⌋()

[constr_1174] `PortInterface` s used in the context of `CompositionSwComponentType` s cannot refer to AUTOSAR services ⌈ `CompositionSwComponentType` s shall not own `PortPrototype` s typed by `PortInterface` s where the attribute `isService` is set to `true` .

⌋()

[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 they shall refer to a `unit` unless the `CompuMethod` 's `category` is one of `TEXTTABLE` , `BITFIELD_TEXTTABLE` , or `IDENTICAL` .

⌋()

[constr_1176] Compatibility of `CompuScale` s of `category` `LINEAR` and `RAT_FUNC` ⌈ `CompuScale` s of `category` `LINEAR` and `RAT_FUNC` are considered compatible if they yield the same conversion.

⌋()

[constr_1177] Allowed `targetCategory` for `SwPointerTargetProps` ⌈ The value of `targetCategory` for `SwPointerTargetProps` can only be one of `TYPE_REFERENCE` or `FUNCTION_REFERENCE` . The only exception from this rule applies if the `swDataDefProps` owned by the `SwPointerTargetProps` refers to a `SwBaseType` with native type declaration `void` , in this case the value `VALUE` is also permitted.

⌋()

[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`

⌋()

[constr_1181] Numerical values used in `ModeDeclaration` . `value` and `ModeDeclarationGroup` . `onTransitionValue` ⌈ The numerical values used to de-

fine the value attributes and the onTransitionValue attribute of a ModeDeclarationGroup shall not overlap.

⌋()

[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).

⌋()

[constr_1183] EndToEndProtectionVariablePrototype s aggregated by EndToEndProtection ⌈ All EndToEndProtectionVariablePrototype s aggregated by the same EndToEndProtection shall refer to the identical sender .

⌋()

[constr_1184] Consistency of rootDataPrototype and base in the context of ApplicationCompositeElementInPortInterfaceInstanceRef ⌈ The rootDataPrototype referenced by ApplicationCompositeElementInPortInterfaceInstanceRef 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 .

⌋()

[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 .

⌋()

[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 .

⌋()

[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 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, with the exception of the shortName of the M1 AutosarDataType .
3. In the context of a DataPrototypeMapping , for each ApplicationCompositeElementDataPrototype of the required DataPrototype a SubElementMapping exists such that a ApplicationCompositeDataTypeSubElementRef 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 .

⌋()

[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` .

⌋()

[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.

⌋()

[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 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 **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 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 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 at the same time (and vice versa).

⌋()

[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.

⌋()

[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 .

⌋()

[constr_1202] Supported connections by AssemblySwConnector for PortPrototype s typed by a SenderReceiverInterface or NvDataInterface ⌈ For the modeling of AssemblySwConnector s between PortPrototype s typed by a SenderReceiverInterface or NvDataInterface , **only** the connections documented in Table 2.12 are supported by AUTOSAR.

⌋()

	RPortPrototype	PPortPrototype	PRPortPrototype
RPortPrototype	No	**Yes**	**Yes**
PPortPrototype	**Yes**	No	**Yes**
PRPortPrototype	**Yes**	**Yes**	**Yes**

Table 2.12: Supported connections for PortPrototype s typed by a SenderReceiverInterface or NvDataInterface

[constr_1203] Supported connections by DelegationSwConnector for PortPrototype s typed by a SenderReceiverInterface or NvDataInterface ⌈ For the modeling of DelegationSwConnector s between PortPrototype s typed by a SenderReceiverInterface or NvDataInterface , **only** the connections documented in Table 2.13 are supported by AUTOSAR.

⌋()

innerPort	outerPort		
	RPortPrototype	PPortPrototype	PRPortPrototype
RPortPrototype	Yes	No	Yes
PPortPrototype	No	Yes	Yes
PRPortPrototype	Yes	Yes	Yes

Table 2.13: Supported connections for PortPrototype s typed by a SenderReceiverInterface or NvDataInterface

[constr_1204] Supported connections by AssemblySwConnector for PortPrototype s typed by a ClientServerInterface , ModeSwitchInterface , or TriggerInterface ⌈ For the modeling of AssemblySwConnector s between PortPrototype s typed by a ClientServerInterface , ModeSwitchInterface , or TriggerInterface , **only** the connections documented in Table 2.14 are supported by AUTOSAR.

⌋()

	RPortPrototype	PPortPrototype	PRPortPrototype
RPortPrototype	No	Yes	Yes
PPortPrototype	Yes	No	No
PRPortPrototype	Yes	No	No

Table 2.14: Supported connections for PortPrototype s typed by a ClientServerInterface , ModeSwitchInterface , or TriggerInterface

[constr_1205] Supported connections by DelegationSwConnector for PortPrototype s typed by a ClientServerInterface , ModeSwitchInterface , or TriggerInterface ⌈ For the modeling of DelegationSwConnector s between PortPrototype s typed by a ClientServerInterface , ModeSwitchInterface , or TriggerInterface , **only** the connections documented in Table 2.15 are supported by AUTOSAR.

⌋()

innerPort	outerPort		
	RPortPrototype	PPortPrototype	PRPortPrototype
RPortPrototype	Yes	No	No
PPortPrototype	No	Yes	No
PRPortPrototype	No	Yes	No

Table 2.15: Supported connections for PortPrototype s typed by a ClientServerInterface , ModeSwitchInterface , or TriggerInterface

[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.

⌋()

[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.

⌋()

[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].

⌋()

[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].

⌋()

[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].

⌋()

[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].

⌋()

[constr_1215] Interpretation of attribute maxNoNewOrRepeatedData owned by EndToEndDescription in PROFILE_01 ⌈ If EndToEndProtection . endToEndProtectionVariablePrototype . receiver is identical to the RPortPrototype . requiredComSpec . dataElement **and** RPortPrototype . requiredComSpec . maxNoNewOrRepeatedData is defined **then** the value of RPortPrototype . requiredComSpec . maxNoNewOrRepeatedData **shall be preferred** over the value of EndToEndProtection . endToEndProfile . maxNoNewOrRepeatedData .

If the value of category of EndToEndDescription is set to PROFILE_01 **and either** the described correspondence rule concerning the referenced VariableDataPrototype is not fulfilled **or** RPortPrototype . requiredComSpec . maxNoNewOrRepeatedData is not defined **then** EndToEndProtection . endToEndProfile . maxNoNewOrRepeatedData **shall exist** .

⌋()

[constr_1216] Interpretation of attribute syncCounterInit owned by EndToEndDescription in PROFILE_01 ⌈ If EndToEndProtection . endToEndPro-

tectionVariablePrototype . receiver is identical to the RPortPrototype . requiredComSpec . dataElement **and** RPortPrototype . requiredComSpec . syncCounterInit is defined **then** the value of RPortPrototype . requiredComSpec . syncCounterInit **shall be preferred** over the value of EndToEndProtection . endToEndProfile . syncCounterInit .

If the value of category of EndToEndDescription is set to PROFILE_01 **and either** the described correspondence rule concerning the referenced VariableDataPrototype is not fulfilled **or** RPortPrototype . requiredComSpec . syncCounterInit is not defined **then** EndToEndProtection . endToEndProfile . syncCounterInit **shall exist** .

⌋()

[constr_1217] Interpretation of attribute maxNoNewOrRepeatedData owned by EndToEndDescription in PROFILE_02 ⌈ If EndToEndProtection . endToEndProtectionVariablePrototype . receiver is identical to the RPortPrototype . requiredComSpec . dataElement **and** RPortPrototype . requiredComSpec . maxNoNewOrRepeatedData is defined **then** the value of RPortPrototype . requiredComSpec . maxNoNewOrRepeatedData **shall be preferred** over the value of EndToEndProtection . endToEndProfile . maxNoNewOrRepeatedData .

If the value of category of EndToEndDescription is set to PROFILE_02 **and either** the described correspondence rule concerning the referenced VariableDataPrototype is not fulfilled **or** RPortPrototype . requiredComSpec . maxNoNewOrRepeatedData is not defined **then** EndToEndProtection . endToEndProfile . maxNoNewOrRepeatedData **shall exist** .

⌋()

[constr_1218] Interpretation of attribute syncCounterInit owned by EndToEndDescription in PROFILE_02 ⌈ If EndToEndProtection . endToEndProtectionVariablePrototype . receiver is identical to the RPortPrototype . requiredComSpec . dataElement **and** RPortPrototype . requiredComSpec . syncCounterInit is defined **then** the value of RPortPrototype . requiredComSpec . syncCounterInit **shall be preferred** over the value of EndToEndProtection . endToEndProfile . syncCounterInit .

If the value of category of EndToEndDescription is set to PROFILE_02 **and either** the described correspondence rule concerning the referenced VariableDataPrototype is not fulfilled **or** RPortPrototype . requiredComSpec . syncCounterInit is not defined **then** EndToEndProtection . endToEndProfile . syncCounterInit **shall exist** .

⌋()

[constr_1219] Invalidation depends on the value of swImplPolicy ⌈ Invalidation of dataElement s is only supported for dataElement s where the value of swImplPolicy is **not** set to queued .

⌋()

[constr_1220] Compatibility of SwBaseType ⌈ Two SwBaseType s are compatible if and only if attributes baseTypeSize respectively byteOrder , memAlignment , baseTypeEncoding , and nativeDeclaration have identical values.

⌋()

[constr_1221] DataPrototype is typed by an ApplicationPrimitiveDataType ⌈ If a DataPrototype is typed by an ApplicationPrimitiveDataType 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.

⌋()

[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 .

⌋()

[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 .

⌋()

[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 or ApplicationRuleBasedValueSpecification .

⌋()

[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 .

⌋()

[constr_1226] Applicable range for ExecutableEntityActivationReason . bitPosition ⌈ The value of attribute ExecutableEntityActivationReason . bitPosition shall be in the range of 0 .. 31.

⌋()

[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 .

⌋()

[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 .

⌋()

[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.

⌋()

[constr_1230] ApplicationDataType that qualifies for Integral Primitive Type ⌈ An ApplicationDataType qualifies as an Integral Primitive Type if and only if **all** of 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]

⌋()

[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 .

⌋()

[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 .

⌋()

[constr_1233] InstantiationTimingEventProps shall only reference TimingEvent ⌈ 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 OperationInvokedEvent shall only be taken from the set of API names associated with the NvM .

⌋()

[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 .

⌋()

[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 .

⌋()

[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 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 .

⌋()

[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 .

⌋()

[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.

⌋()

[constr_1244] DataPrototype s used in application software shall not be typed by C enums ⌈ A DataPrototype that is used in an AtomicSwComponentType shall not set swDataDefProps . additionalNativeTypeQualifier to enum .

⌋()

[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:

- **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`

⌋()

[constr_1247] Consistency of `ModeDeclarationMappingSet` with respect to the referenced `firstModeGroup` and `secondModeGroup` ⌈ If a `ModeDeclarationGroupPrototypeMapping` . `modeDeclarationMappingSet` exists, 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` .

⌋()

[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 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 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 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 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 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 ⌈ 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] Supported usage of VariationPointProxy ⌈ The allowed multiplicities for attributes of VariationPointProxy depending on the applicable binding time and the value of VariationPointProxy . category are documented in Table **??** .

For clarification, the multiplicities of attributes of meta-class VariationPointProxy that are **not** explicitly mentioned in a given row of table **??** shall be interpreted as [0].

⌋()

[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 .

⌋()

[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.

⌋()

[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 .

⌋()

[constr_1257] No WaitPoint s allowed ⌈ A RunnableEntity referenced by an InitEvent in the role startOnEvent shall not aggregate a WaitPoint .

⌋()

[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.

⌋()

[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 .

⌋()

[constr_1260] No mode disabling for `InitEvent` s ⌈ An `InitEvent` shall not have a reference to a `ModeDeclaration` in the role `disabledMode` .

⌋()

[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.

⌋()

[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` .

⌋()

[constr_1264] Iteration along output axis is only supported for `VALUE` and `VAL_BLK` ⌈ `swRecordLayoutVIndex` in `SwRecordLayoutV` cannot be 0 for any value of `SwRecordLayoutV` . `category` other than `VALUE` and `VAL_BLK` .

⌋()

[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.

⌋()

[constr_1269] Number of `argument` s shall be preserved in a `ClientServerOperationMapping` ⌈ Within the context of a `ClientServerOperationMapping` , the number of `argument` s of `firstOperation` and `secondOperation` shall be identical.

⌋()

[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` .

⌋()

[constr_1271] `RecordValueSpecification` . `field` s shall be identical to the number of `ApplicationRecordDataType` . `element` s ⌈ The initialization of an `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.

⌋()

[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.

⌋()

[constr_1273] Rules for the initialization of ApplicationArrayDataType by means of ArrayValueSpecification ⌈ The following rules apply for the initialization 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 has element s** 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 does not have any element s** then this configuration shall be accepted as an “empty initialization”, see [TPS_SWCT_01752] .

⌋()

[constr_1274] Rules for the initialization of array-shaped ImplementationDataType by means of ArrayValueSpecification ⌈ The following rules apply for the initialization of a DataPrototype typed by an ImplementationDataType of category ARRAY by means of an ArrayValueSpecification :

- If the attribute ImplementationDataType . subElement . arraySizeSemantics is set to **fixedSize** then 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 .

- If the attribute ImplementationDataType . subElement . arraySizeSemantics is set to **variableSize** and the **ArrayValueSpecification has element s** then 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 .

- If the attribute ImplementationDataType . subElement . arraySizeSemantics is set to **variableSize** and the **ArrayValueSpecification does not have any element s** then this configuration shall be accepted as an "empty initialization", see [TPS_SWCT_01752] .

⌋()

[constr_1277] SwDataDefProps . swImplPolicy of a VariableDataPrototype referenced by a VariableAccess aggregated in the role dataReceivePointByValue ⌈ The SwDataDefProps . swImplPolicy of a VariableDataPrototype referenced by a VariableAccess aggregated in the role dataReceivePointByValue shall not be set to queued .

⌋()

[constr_1278] PhysConstrs references a Unit ⌈ DataConstr s are only compatible if the DataConstr . dataConstrRule . physConstrs . unit are compatible or neither DataConstr . dataConstrRule . physConstrs . unit exist.

⌋()

[constr_1279] Unmapped elements of ApplicationCompositeDataType s or ImplementationDataType s and the attribute swImplPolicy ⌈ If the attribute swImplPolicy is set to queued it is not allowed to have unmapped elements of ApplicationCompositeDataType s or ImplementationDataType s of category STRUCTURE or ARRAY on the receiver side.

⌋()

[constr_1280] Unmapped dataElement on the receiver side shall have an init Value ⌈ If elements of ApplicationCompositeDataType s or ImplementationDataType s of category STRUCTURE or ARRAY are not considered in a SubElementMapping then the enclosing dataElement shall have an initValue **if** the NonqueuedReceiverComSpec is aggregated by an AbstractRequiredPortPrototype .

⌋()

[constr_1281] `invalidValue` is inside the scope of the `compuMethod` ⌈ If the value of the `invalidValue` of an `ApplicationPrimitiveDataType` of `category` `VALUE` is supposed to be **inside** the scope of the applicable `CompuMethod` an `ApplicationValueSpecification` is used to describe the `invalidValue` of the `ApplicationPrimitiveDataType` .

⌋()

[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.

⌋()

[constr_1283] `invalidValue` is outside the scope of the `compuMethod` ⌈ If the value of the `invalidValue` of an `ApplicationPrimitiveDataType` of `category` `VALUE` is supposed to be **outside** the scope of the applicable `CompuMethod` a `NumericalValueSpecification` shall be used to describe the `invalidValue` of the `ApplicationPrimitiveDataType` .

⌋()

[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` .

⌋()

[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 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 `Argu-`

mentDataPrototype of direction in that is typed by an AutosarDataType that boils down to a primitive C data type (see [TPS_SWCT_01565]).

⌋()

[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.

⌋()

[constr_1288] Allowed Attributes vs. category for DataPrototype s typed by ImplementationDataType s ⌈ The allowed values per category for DataPrototype s typed by ImplementationDataType s are documented in table 2.16 .

⌋()

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 [18]			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										

▽

[18] don't care

△

Attributes of SwDataDefProps	Root Element			Attribute Existence per Category						
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										

Table 2.16: Allowed Attributes vs. category for DataPrototype s typed by ImplementationDataType s

[constr_1289] Allowed Attributes vs. category for DataPrototype s typed by ApplicationDataType s ⌈ The allowed values of Attributes per category for DataPrototype s typed by ApplicationDataType s are documented in table 2.17 .

⌋()

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 [19]	d/c		d/c		d/c			d/c	d/c	d/c	d/c	d/c

▽

[19] don't care

△

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
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																
swCalprmAxisSet . swCalprmAxis . baseType																
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																
swRecordLayout																
swRefreshTiming	x	x		0..1	0..1			0..1	0..1							
swTextProps																
swValueBlockSize																
swValueBlockSizeMult																
unit																
valueAxisDataType																

Table 2.17: Allowed Attributes vs. category for DataPrototype s typed by Application Data Types

[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** PPortComSpec that references a given dataElement or operation .

⌋()

[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 or operation .

⌋()

[constr_1292] Limitation on the number of RPortComSpec s/ PPortComSpec s 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 or operation .

⌋()

[constr_1295] PortInterface s 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 .

⌋()

[constr_1296] DataPrototype s 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 .

⌋()

[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 .

⌋()

[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 .

⌋()

[constr_1300] Primitive DataPrototype on the provider side shall not be mapped to element of a composite data type on the requester side ⌈ The usage of DataPrototypeMapping or SubElementMapping does not support the following configuration:

- The AutosarDataPrototype referenced on the provider/client side 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 on the requester/server side.

⌋()

[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 .

⌋()

[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 ApplicationPrimitiveDataType 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 ApplicationCompositeDataType 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 .

⌋()

[constr_1303] Applicability of TextTableMapping depending on the value of CompuMethod . category ⌈ If 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 .

⌋()

[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 .

⌋()

[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.

⌋()

[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.

⌋()

[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 .

⌋()

[constr_1309] Existence of NvBlockDescriptor . timingEvent ⌈ The attribute NvBlockDescriptor . timingEvent shall exist 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

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 .

⌋()

[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:

- **safetyQM**
- **safetyAsilA**
- **safetyAsilB**
- **safetyAsilC**
- **safetyAsilD**

⌋()

[constr_1312] PortPrototype s typed by a ParameterInterface ⌈ PortPrototype s typed by a ParameterInterface can either be PPortPrototype s or RPortPrototype s. The usage of PRPortPrototype s that are typed by a ParameterInterface is not supported.

⌋()

[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** [20] is represented by exactly one `TextTableValuePair` . `firstValue` ([TPS_SWCT_01163]) or `TextTableValuePair` . `secondValue` ([TPS_SWCT_01164]).

⌋()

[constr_1314] Profile `VSA_LINEAR` for `ApplicationArrayDataType` ⌈ If the `dynamicArraySizeProfile` of `ApplicationArrayDataType` is set to `VSA_LINEAR` , the contained `ApplicationArrayElement` shall fulfill **all** of 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 `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** of 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` .

[20] 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] .

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** of 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** of 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 .

⌋()

[constr_1316] Profile VSA_RECTANGULAR for ApplicationArrayDataType ⌈
If the dynamicArraySizeProfile of ApplicationArrayDataType is set to VSA_RECTANGULAR the contained ApplicationArrayElement shall fulfill **all** of 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** of 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** of 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 .

⌋()

[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** of 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 ApplicationArrayElement that fulfills **all** of 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** of 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 .

⌋()

[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 of 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 of 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 .

⌋()

[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 of the 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 of 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 of 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 of 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 .

⌋()

[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 of 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 of 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** ImplementationDataTypeElement s in the aggregation chain that do not terminate the chain shall fulfill all of 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 of 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 .

⌋()

[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 of 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 of 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 of 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 of 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 of 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 .

⌋()

[constr_1322] Size Indicator for undefined dynamicArraySizeProfile ⌈ If the ImplementationDataType . dynamicArraySizeProfile does not exists 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.

⌋()

[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.

⌋()

[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.

⌋()

[constr_1375] Existence of attributes of `CompuMethod` and related meta-classes ⌈ The existence of attributes of `CompuMethod` and related meta-classes depending on the value of the `category` shall follow the restrictions documented in Table 2.18 .

⌋()

	Attribute Existence per 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
`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/1 v	D/1 v	D	D	N/A	N/A	D/1 v	D	N/A
`CompuRationalCoeffs.compuNumerator`	N/A	D/2 v	D/2 v	D	D	N/A	N/A	D/2 v	D	N/A

Table 2.18: Allowed Attributes vs. `category` for `CompuMethod` s

[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:

- **`coreGlobal`**
- **`coreLocal`**

⌋()

[constr_1382] Mutually exclusive existence of attributes `SwVariableRefProxy.autosarVariable` vs. `SwVariableRefProxy.mcDataInstanceVar` ⌈ In any given AUTOSAR model, the aggregations `SwVariableRefProxy.autosarVari-`

able and SwVariableRefProxy . mcDataInstanceVar shall never exist at the same time.

⌋()

[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 .

⌋()

[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

⌋()

[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 .

⌋()

[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 .

⌋()

[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.

⌋()

[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 .

⌋()

[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 .

⌋()

[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 .

⌋()

[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 .

⌋()

[constr_1393] Existence of RuleBasedValueCont . unit ⌈ For every RuleBasedValueCont the attribute unit shall exist.

⌋()

[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.

⌋()

[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 .

⌋()

[constr_1397] Existence of attributes of `TransformerHardErrorEvent` ⌈ For any given `TransformerHardErrorEvent` , **either** the attribute `TransformerHardErrorEvent . operation` **or** `TransformerHardErrorEvent . trigger` shall exist.

⌋()

[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`

⌋()

[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` .

⌋()

[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`

⌋()

[constr_1401] Restrictions on the relation between `DataPrototypeMapping` and `DataTransformation` ⌈ A `VariableDataPrototype` in the context of a `PortPrototype` shall **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 .

⌋()

[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.

⌋()

[constr_1404] All NvDataInterface . nvData of PortPrototype s in the context of a specific SwcServiceDependency shall be mapped to the same NvBlockDescriptor ⌈ In the context of a given SwcServiceDependency (which, in turn, is owned by an AtomicSwComponentType), **all** NvDataInterface . nvData of PortPrototype s referenced by a RoleBasedPortAssignment with attribute RoleBasedPortAssignment . role set to NvDataPort shall be connected (either directly or via the definition of suitable PortInterfaceMapping s) to NvDataInterface . nvData (on the side of the NvBlockSwComponentType) that are **completely mapped** (via NvBlockDataMapping s) **to the identical** NvBlockDescriptor . ramBlock .

⌋()

[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 .

⌋()

[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 .

⌋()

[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 Ap-

plicationArrayDataType or an ImplementationDataType of category ARRAY if the aggregated ApplicationArrayDataType . element or ImplementationDataType . subElement also supports the specification of a SwDataDefProps . dataConstr .

⌋()

[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 .

⌋()

[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 .

⌋()

[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 .

⌋()

[constr_1415] Supported values of ModeSwitchEventTriggeredActivity . role ⌈ The only supported value of ModeSwitchEventTriggeredActivity . role is WriteBlock .

⌋()

[constr_1416] Existence of ApplicationArrayElement . maxNumberOfElements ⌈ The attribute ApplicationArrayElement . maxNumberOfElements shall exist 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.

⌋()

[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.

⌋()

[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.

⌋()

[constr_1422] Value of `category` is `VOID` ⌈ If the value of the attribute `SwBaseType` . `category` is set to `VOID` then the attribute `baseTypeSize` shall not exist.

⌋()

[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 `ImplementationDataType` or `ImplementationDataTypeElement` of `category` `ARRAY` in a chain of referencing `ImplementationDataType` s

starting from the `ImplementationDataType` s of the `rootVariableDataPrototype` down to the leaf `ImplementationDataTypeElement` which is typed (directly

or indirectly via ImplementationDataType of category TYPE_REFERENCE) by the ImplementationDataType of the targetDataPrototype .

⌋()

[constr_1424] Existence of ArVariableInImplementationDataInstanceRef . contextDataPrototype ⌈ The attribute ArVariableInImplementationDataInstanceRef . contextDataPrototype shall only exist for an ImplementationDataTypeElement category TYPE_REFERENCE or ARRAY .

⌋()

[constr_1425] Definition of swCalprmAxisSet . swCalprmAxis / SwAxisIndividual . swVariableRef depending on the capabilities of the data type ⌈ The definition of a swCalprmAxisSet . swCalprmAxis / SwAxisIndividual . swVariableRef in the context of an InstantiationDataDefProps or a ParameterAccess is only supported for a DataPrototype of category ARRAY if the data type of the ApplicationArrayElement also supports the specification of a swCalprmAxisSet . swCalprmAxis / SwAxisIndividual . swVariableRef according to [constr_1289] .

Thereby, multiple ApplicationArrayDataType s might be nested to express multiple array dimensions.

⌋()

[constr_1426] Consistency of array sizes for axes and input variable array ⌈ The number of array dimension defined by ApplicationArrayDataType s and the values 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 **identical** to the number of array dimension and according value of the maxNumberOfElements of the VariableDataPrototype referenced by SwAxisIndividual . swVariableRef . autosarVariable .

⌋()

[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 ApplicationArrayElement 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.

⌋()

[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 .

⌋()

[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 .

⌋()

[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.

⌋()

[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 .

⌋()

[constr_1432] Multiplicity of CommunicationBufferLocking ⌈ In a concrete aggregated set of PortAPIOption . supportedFeature , CommunicationBufferLocking shall exist **at most once** .

⌋()

[constr_1433] Transient faults are not applicable to software-components ⌈ An ErrorTracerNeeds aggregated in the context of a SwcInternalBehavior is not allowed to own a TransientFault in the role ErrorTracerNeeds . tracedFailure .

⌋()

[constr_1434] CompuScale s shall not have identical CompuScale Value Symbolic Name s ⌈ In a CompuMethod that is subject to [constr_1146] , no two Com-

puScale s shall have identical CompuScale Value Symbolic Name s (according to [TPS_SWCT_01696]).

⌋()

[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 .

⌋()

[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.

⌋()

[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.

⌋()

[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 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 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** .

⌋()

[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.

⌋()

[constr_1468] Limitation on the number of `SwcExclusiveAreaPolicy` s ⌈ An `ExclusiveArea` shall only be referenced by **at most** one `SwcExclusiveAreaPolicy` .

⌋()

[constr_1469] Applicability of constraints depending on the existence of a data transformation ⌈ [constr_1269] , [constr_1270] , [constr_1268] , and [constr_1240] shall **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` .

⌋()

[constr_1517] Existence of `ArParameterInImplementationDataInstanceRef` . `contextDataPrototype` ⌈ The attribute `ArParameterInImplementationDataInstanceRef` . `contextDataPrototype` shall only exist 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` .

⌋()

[constr_1519] Existence of attributes vs. `category` of `ApplicationValueSpecification` ⌈ The existence of attributes of meta-class `ApplicationValueSpecification` vs. the value of `category` is regulated by Table 2.19 .

⌋()

	Attribute Existence per Category									
Attribute of `ApplicationValueSpecification`	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)

Table 2.19: Allowed Attributes vs. `category` for `ApplicationValueSpecification`

[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` .

⌋()

[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` .

⌋()

[constr_1523] No mode disabling for `OperationInvokedEvent` s ⌈ An `OperationInvokedEvent` shall not have a reference to a `ModeDeclaration` in the role `disabledMode` .

⌋()

[constr_1538] Restriction for `ReceiverComSpec` . `dataElement` ⌈ The reference `ReceiverComSpec` . `dataElement` **shall not** refer to an `ArgumentDataPrototype` or `ParameterDataPrototype` .

⌋()

[constr_1539] Restriction for SenderComSpec . dataElement ⌈ The reference SenderComSpec . dataElement **shall not** refer to an ArgumentDataPrototype or ParameterDataPrototype .

⌋()

[constr_1540] Existence of ClientComSpec . operation ⌈ The reference ClientComSpec . operation **shall exist** if the AbstractRequiredPortPrototype that owns the ClientComSpec is typed by a ClientServerInterface .

⌋()

[constr_1541] Existence of ServerComSpec . operation ⌈ The reference ServerComSpec . operation **shall exist** if the AbstractProvidedPortPrototype that owns the ServerComSpec is typed by a ClientServerInterface .

⌋()

[constr_1544] Modeling of SwAxisGeneric for the definition of a fix axis ⌈ The standardized values and multiplicities within the model of an SwAxisGeneric according to [TPS_SWCT_01479] and [TPS_SWCT_01480] are documented in Table 2.20 .

⌋()

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..*

Table 2.20: Modeling of SwAxisGeneric

[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 .

⌋()

[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** .

⌋()

[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 ApplicationArrayDataType or ImplementationDataType of category ARRAY supports the specification of a SwDataDefProps . displayPresentation .

⌋()

[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 .

⌋()

[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 [1441] .

⌋()

[constr_1608] Existence of rootParameterDataPrototype ⌈ The reference rootParameterDataPrototype shall exist 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 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 SwDataDefProps . swValueBlockSizeMult ⌈ Attributes SwDataDefProps . swValueBlockSize and SwDataDefProps . swValueBlockSizeMult shall not exist at the same time in the context of a given SwDataDefProps .

⌋()

[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.

⌋()

[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 .

⌋()

[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 .

⌋()

[constr_1632] Restriction for firstToSecondDataTransformation and secondToFirstDataTransformation ⌈ If 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 ⌈ the allowed combinations of ApplicationDataType . category vs. CompuMethod . category are described by Table 2.21 .

⌋()

	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

Table 2.21: ApplicationDataType . category vs. CompuMethod . category

[constr_1635] Relevance of attribute isOptional ⌈ If a SubElementMapping is defined for the elements of a structured data type then the attribute isOptional [21] shall either not exist for the firstElement and secondElement or it shall have the identical value for the firstElement and secondElement .

⌋()

[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 Element Structure (see [TPS_SWCT_01774]).

⌋()

[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 .

⌋()

[constr_1638] First ImplementationDataTypeElement of ImplementationDataType that represents an Optional Element Structure ⌈ The first ImplementationDataTypeElement of ImplementationDataType that represents

[21] this is valid for both ApplicationRecordElement as well as ImplementationDataTypeElement

an Optional Element Structure , i.e. the availabilityBitfield according to [TPS_SWCT_01774] , shall not set attribute isOptional to True .

⌋()

[constr_1639] ImplementationDataTypeElement with attribute isOptional set to True ⌈ ImplementationDataTypeElement with attribute isOptional set to True shall **not** be of category STRUCTURE .

⌋()

[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 .

⌋()

[constr_1662] Compatibility of ApplicationRecordDataType and ImplementationDataType that both represent an Optional Element Structure ⌈ An ApplicationRecordDataType that represents an Optional Element Structure can 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 .

⌋()

[constr_2000] Compatibility of ClientServerOperation s triggering the same RunnableEntity ⌈ The ClientServerOperation s are considered compatible if the number of arguments (which can be ArgumentDataPrototype s or related PortDefinedArgumentValue s) is equal and the corresponding arguments (i.e. first argument on both sides, second argument on both sides, etc.) are compatible.

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 .

⌋()

[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 .

⌋()

[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 .

⌋()

[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 .

⌋()

[constr_2006] Number of AsynchronousServerCallResultPoint referencing to one AsynchronousServerCallPoint ⌈ The AsynchronousServerCallPoint may be be referenced by at most one AsynchronousServerCallResultPoint .

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.

⌋()

[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

⌋()

[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.

⌋()

[constr_2011] Connections between `SwComponentPrototype` s typed by `NvBlockSwComponentType` and `SwComponentPrototype` s typed by other `AtomicSwComponentType` s ⌈ The *nv data* `PortPrototype` s of the `SwComponentPrototype` typed by an `NvBlockSwComponentType` are either connected with `PortPrototype` s typed by `NvDataInterface` s or `SenderReceiverInterface` s of other `AtomicSwComponentType` .

⌋()

[constr_2012] Compatibility of `ImplementationDataType` s used for `ramBlock` and `romBlock` ⌈ The `ramBlock` and the `romBlock` shall have compatible `ImplementationDataType` s to ensure, 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` ⌈ 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 `VariableDataPrototype` aggregated by `NvBlockDataMapping` in the role `writtenNvData` , `writtenReadNvData` , or `readNvData` .

⌋()

[constr_2014] Limitation of `RoleBasedPortAssignment` . `role` in `NvBlockDescriptor` s ⌈ The `role` has to be set to a valid name of the *Standardized AUTOSAR*

Interface used for the *NVRAM Manager* e.g. *NvMNotifyJobFinished* or *NvMNotifyInit Block* .

⌋()

[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

⌋()

[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.

⌋()

[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.

⌋()

[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.

⌋()

[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 . The exceptions described in [TPS_SWCT_01572] , [TPS_SWCT_01579] 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 .

⌋()

[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 .

⌋()

[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.

⌋()

[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.

⌋()

[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 .

⌋()

[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 .

⌋()

[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` .

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_01580] , as well as [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`

⌋()

[constr_2029] `shortName` of `constantMemory` and `staticMemory` ⌈ The `shortName` of a `VariableDataPrototype` in role `staticMemory` or a `ParameterDataPrototype` in role `constantMemory` has to be equal with the 'C' identifier of the described variable or constant.

⌋()

[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` .

⌋()

[constr_2031] Period of `TimingEvent` shall be greater than 0 ⌈ The value of the attribute period of `TimingEvent` shall be greater than 0.

⌋()

[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 .

⌋()

[constr_2034] SwAddrMethod referenced by RunnableEntity s or BswSchedulableEntity s ⌈ RunnableEntity s and BswSchedulableEntity s shall not reference a SwAddrMethod which attribute memoryAllocationKeywordPolicy is set to addrMethodShortNameAndAlignment .

⌋()

[constr_2035] swImplPolicy for VariableDataPrototype in SenderReceiverInterface ⌈ The overriding swImplPolicy attribute value of a VariableDataPrototype in SenderReceiverInterface shall be standard , queued or measurementPoint .

⌋()

[constr_2036] swImplPolicy for VariableDataPrototype in NvDataInterface ⌈ The overriding swImplPolicy attribute value of a VariableDataPrototype in NvDataInterface shall be standard .

⌋()

[constr_2037] swImplPolicy for VariableDataPrototype in the role ramBlock ⌈ The overriding swImplPolicy attribute value of a VariableDataPrototype in the role ramBlock shall be standard .

⌋()

[constr_2038] swImplPolicy for VariableDataPrototype in the role implicitInterRunnableVariable ⌈ The overriding swImplPolicy attribute value of a VariableDataPrototype in the role implicitInterRunnableVariable shall be standard .

⌋()

[constr_2039] swImplPolicy for VariableDataPrototype in the role explicitInterRunnableVariable ⌈ The overriding swImplPolicy attribute value of a VariableDataPrototype in the role explicitInterRunnableVariable shall be standard .

⌋()

[constr_2040] swImplPolicy for VariableDataPrototype in the role arTypedPerInstanceMemory ⌈ The overriding swImplPolicy attribute value of a VariableDataPrototype in the role arTypedPerInstanceMemory shall be standard or measurementPoint .

⌋()

[constr_2041] **swImplPolicy for VariableDataPrototype in the role staticMemory** ⌈ The overriding swImplPolicy attribute value of a VariableDataPrototype in the role staticMemory shall be standard or measurementPoint . ⌋()

[constr_2042] **swImplPolicy for ParameterDataPrototype in ParameterInterface** ⌈ The overriding swImplPolicy attribute value of a ParameterDataPrototype in ParameterInterface shall be standard , const or fixed . ⌋()

[constr_2043] **swImplPolicy for ParameterDataPrototype in the role romBlock** ⌈ The overriding swImplPolicy attribute value of a ParameterDataPrototype in the role romBlock shall be standard . ⌋()

[constr_2044] **swImplPolicy for ParameterDataPrototype in the role sharedParameter** ⌈ The overriding swImplPolicy attribute value of a ParameterDataPrototype in the role sharedParameter shall be standard , const . ⌋()

[constr_2045] **swImplPolicy for ParameterDataPrototype in the role perInstanceParameter** ⌈ The overriding swImplPolicy attribute value of a ParameterDataPrototype in the role perInstanceParameter shall be standard , const . ⌋()

[constr_2046] **swImplPolicy for ParameterDataPrototype in the role constantMemory** ⌈ The overriding swImplPolicy attribute value of a ParameterDataPrototype in the role constantMemory shall be standard , const or fixed . ⌋()

[constr_2047] **swImplPolicy for ArgumentDataPrototype** ⌈ The overriding swImplPolicy attribute value of a ArgumentDataPrototype shall be standard . ⌋()

[constr_2048] **swImplPolicy for SwServiceArg** ⌈ The overriding swImplPolicy attribute value of a SwServiceArg shall be standard or const . ⌋()

[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.

⌋()

[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.

⌋()

[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 .

⌋()

[constr_2052] Values of swArraysize and the number of values provided by swValuesPhys shall be consistent. ⌈ swValuesPhys shall define as many numbers of values as the swArraysize 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 these have to be multiplied in order to get the total number of swValuesPhys values.

⌋()

[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.

⌋()

[constr_2054] Valid targets of rptSystem ⌈ The System referenced in the role rptSystem shall be of category RPT_SYSTEM .

⌋()

[constr_2055] Valid targets of byPassPoint and rptHook reference ⌈ Depending on the category value the targets of byPassPoint and rptHook references are restricted according table 2.22 .

⌋()

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.

Table 2.22: Category of RptContainers

[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 .

⌋()

[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 .

⌋()

[constr_2058] Mandatory information of a RuleBasedValueCont ⌈ If the attribute swValueCont is defined for an ApplicationRuleBasedValueSpecification the RuleBasedValueCont shall define always the attribute swArraysize if the ApplicationRuleBasedValueSpecification is of category CURVE , MAP , CUBOID , CUBE_4 , CUBE_5 , COM_AXIS , RES_AXIS , VAL_BLK or ARRAY .

⌋()

[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 .

⌋()

[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` .

⌋()

[constr_2544] Limits need to be consistent ⌈

- The limits of `ApplicationDataType` shall be inside of 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` .

⌋()

[constr_2545] `invalidValue` shall fit in the specified ranges ⌈ The `invalidValue` shall be in the range of the `ImplementationDataType` .

⌋()

[constr_2548] Data constraint of value axis shall match ⌈ The values compliant to `SwDataDefProps` . `dataConstr` shall be 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

⌋()

[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.

⌋()

[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.

⌋()

[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.

Other values may exist, but the semantics is outside of the AUTOSAR scope.

⌋()

[constr_4000] Local communication of mode switches ⌈ Ports with ModeSwitchInterface s cannot be connected across ECU boundaries.

⌋()

[constr_4002] Unambiguous mapping of modes to data types ⌈ Within one DataTypeMappingSet , a ModeDeclarationGroup shall not be mapped to different ImplementationDataType s.

⌋()

[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 .

⌋()

[constr_4004] Context of SenderReceiverAnnotation ⌈ A SenderReceiverAnnotation shall only be aggregated by a PortPrototype typed by a SenderReceiverInterface .

⌋()

[constr_4005] Context of ClientServerAnnotation ⌈ A ClientServerAnnotation shall only be aggregated by a PortPrototype typed by a ClientServerInterface .

⌋()

[constr_4006] Context of ParameterPortAnnotation ⌈ A ParameterPortAnnotation shall only be aggregated by a PPortPrototype owned by a ParameterSwComponentType .

⌋()

[constr_4007] Context of ModePortAnnotation ⌈ A ModePortAnnotation shall only be aggregated by a PortPrototype typed by a ModeSwitchInterface .

⌋()

[constr_4008] Context of `TriggerPortAnnotation` ⌈ A `TriggerPortAnnotation` shall only be aggregated by a `PortPrototype` typed by a `TriggerInterface` .

⌋()

[constr_4009] Context of `NvDataPortAnnotation` ⌈ An `NvDataPortAnnotation` shall only be aggregated by a `PortPrototype` typed by an `NvDataInterface` .

⌋()

[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` .

⌋()

[constr_4035] `ValueSpecification` shall fit into data type ⌈ An instance of `ValueSpecification` which is used to assign a value to a software object typed by an `AutosarDataType` shall fit into this `AutosarDataType` without losing information.

⌋()

[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.

⌋()

2.33 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 [10] , 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

SPECIFICATION_ITEM The text represents a particular item in the specification. Such an item is a requirement for the implementation of the software specification.

REQUIREMENT_ITEM The text represents a particular requirement. Such an item is applicable primarily in requirement specifications.

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.

TEST_ITEM The text represents a short description of a test. Such an item is applicable primarily in test specifications.

SAFETY_* The text represents the type of safety requirements. The allowed values (*) are defined in [TPS_SAFEX_00102] in [11] .

⌋()

[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 `blueprintCondition` 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 [10] is relaxed for blueprints. This means in particular, that all `PackageableElement` s which inherit from `AtpBlueprint` and live in a package of category BLUEPRINT may have a `VariationPoint` . In this case `vh.latestBindingTime` is considered as `blueprintDerivationTime` 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` 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 [12] . For navigation over the model we use the names as they are used in XML.

⌋()

[constr_2597] `ClientServerOperationBlueprintMapping` constraints number of arguments ⌈ The number of arguments of the `BswModuleEntry` referenced by a `bswModuleEntry` shall be identical to the number of portDefinedArgumentBlueprints of the owning `ClientServerInterfaceToBswModuleEntryBlueprintMapping` plus the number of `ArgumentDataPrototype` s aggregated in the role argument of the `clientServerOperation`

⌋()

[constr_2598] `ClientServerOperationBlueprintMapping` constraints 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 PrimitiveAttributeTailoring 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 ⌈ Table 2.23 defines the allowed combinations of uptraces with respect to life cycle states [TPS_STDT_00064] .

⌋()

	Trace to					
Trace from	draft	valid	obsolete	preliminary	removed	shall Become Mandatory
draft	1	1	0	1	0	1
valid	0	1	0	0	0	0
obsolete	1	1	1	1	0	1
preliminary	1	1	0	1	0	1
removed	1	1	1	1	1	1
shallBecomeMandatory	0	1	0	0	0	1

Table 2.23: Matrix of allowed uptraces wrt. life cycles

2.34 TPS_SystemTemplate

[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 the context of a `TriggerToSignalMapping`, it is only possible to refer to a `TriggerToSignalMapping.systemSignal` that in turn is referenced by an `ISignal` with attribute `length` set to 0.

⌋()

[constr_1199] `ISignal` s relating to `systemSignal` s eligible for a `TriggerToSignalMapping` ⌈ An `ISignal` used to reference a `systemSignal` that in turn is referenced by a `TriggerToSignalMapping` shall also be referenced by an `ISignalToIPduMapping` where the attribute `updateIndicationBitPosition` is defined.

⌋()

[constr_1207] Existence of the attribute `DataMapping` . `communicationDirection` in the context of a `SenderReceiverInterface` or `TriggerInterface` ⌈ The following condition shall be fulfilled regarding the existence and values of the attribute `DataMapping` . `communicationDirection` that refers to a `PortPrototype` typed by a `SenderReceiverInterface` or `TriggerInterface` as the context `PortPrototype` :

- If the `DataMapping` refers to a `PRPortPrototype` as the context `PortPrototype` the attribute `DataMapping` . `communicationDirection` shall exist.
- If the `DataMapping` refers to a `PPortPrototype` as the context `PortPrototype` the attribute `DataMapping` . `communicationDirection` may exist. If the attribute exists its value shall be set to `out` .
- If the `DataMapping` refers to an `RPortPrototype` as the context `PortPrototype` the attribute `DataMapping` . `communicationDirection` may exist. If the attribute exists its value shall be set to `in` .

⌋()

[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 `GlobalTimeGateway` 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_01642] and [TPS_SWCT_01643] 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 `ImplementationDataType` 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 `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 `ApplicationRecordDataType` ⌈ 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 `ImplementationDataType` or `ImplementationDataTypeElement` ⌈ For every `ImplementationDataType` or `ImplementationDataTypeElement` of `category` `STRUCTURE` where direct members set the attribute `ImplementationDataTypeElement` . `isOptional` to the value `True` references to all direct members of this `ImplementationDataType` 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 `TlvDataIdDefinition` . `tlvArgument` s shall exist for all `argument` s of the given

ClientServerOperation and all affected TlvDataIdDefinition shall be aggregated by the same SOMEIPTransformationISignalProps .

⌋()

[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 shall be aggregated by the same SOMEIPTransformationISignalProps .

⌋()

[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 shall be aggregated by the same SOMEIPTransformationISignalProps .

⌋()

[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 TlvDataIdDefinition . 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` , 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` , 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` in the *server* role ⌈ **Either of** the references to

- `CryptoServiceCertificate` in the role `TlsCryptoCipherSuite` . `certificate`
- `CryptoServiceKey` in the role `TlsCryptoCipherSuite` . `preSharedKey`

shall exist if the CryptoServiceCertificate is referenced from a TlsCryptoServiceMapping that has attribute category set to the value TLS_SERVER .

⌋()

[constr_2025] Uniqueness of symbol attributes ⌈ With the exception of Runnable Entities that are subject to 1234 (RunnableEntities owned by NvBlockSwComponent Types), 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 ⌈ SenderReceiverToSignalGroupMapping . signalGroup . systemSignal shall point to each SystemSignal being mapped within the context of SenderReceiverToSignalGroupMapping .

In other words: For each SystemSignal referenced in the role SenderReceiverToSignalGroupMapping . signalGroup . systemSignal there shall be either a reference in the role SenderRecRecordElementMapping . systemSignal or a reference in the role SenderRecArrayElementMapping . systemSignal aggregated by the same SenderReceiverToSignalGroupMapping that refers to this SystemSignal .

⌋()

[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 must be exclusive ⌈ Clustering and separation must be exclusive, i.e. it SHALL NOT be possible that two SwComponentPrototype s A and B are associated by a ComponentClustering and by a ComponentSeparation .

⌋()

[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` must 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 `Frame` shall NOT overlap.

⌋()

[constr_3013] Frame length shall not be exceeded ⌈ The combined length of all Pdu s that are mapped into a Frame shall not exceed the defined Frame length.

⌋()

[constr_3014] Overlapping of updateIndicationBits for Pdu s is prohibited ⌈ The updateIndicationBitPosition for a Pdu in a Frame 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 must 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 must 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 SensorActuatorSwComponentType shall be mapped to a HwElement with category Sensor Actuator via the controlledHwElement relation.

⌋()

[constr_3025] Usage of NPdu s in TpConnection s ⌈ In case several TpConnection 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_DESCRIPTION` 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_3035] CanNm user data configuration in case NID/CBV are enabled ⌈ If NID/CBV are enabled (`nmCbvPosition` and `nmNidPosition` are configured), there shall not be any user data configured at the position of the respective NID/CBV bytes.

⌋()

[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.

⌋()

[constr_3040] Restriction of `pncIdentifier` values ⌈ The `pncIdentifier` value shall be within the range described by `pncVectorOffset` and `pncVectorLength` .

⌋()

[constr_3041] `pncVectorOffset` range ⌈ The `pncVectorOffset` value shall be in the range of 1..7.

⌋()

[constr_3042] `pncVectorLength` range ⌈ The `pncVectorLength` value shall be in the range of 1..7.

⌋()

[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_3049] Role of `SystemSignal` in inter-ECU client server communication with clients located on different ECUs in case of networks other than Ethernet ⌈ In case of a n:1 inter-ECU client server communication with clients located on different ECUs different `SystemSignal` s shall be used for each Ecu.

⌋()

[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_3055] `SystemSignalGroup` in a complete System Description ⌈ For each `SystemSignalGroup` in a complete `System` with `category` `SYSTEM_DESCRIPTION` exactly one `DataMapping` shall be defined (`PPortPrototype` or `RPortPrototype`). Preference: `PPortPrototype`

⌋()

[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_3059] Mandatory `DataMapping` on the receiver side for elements of a composite data type ⌈ On the receiver side, it is required that for every `ApplicationCompositeElementDataPrototype` of an `ApplicationCompositeDataType` (`ApplicationArrayDataType` . `element` resp. `Application-`

RecordDataType . element) that types a dataElement in a RPortPrototype or PRPortPrototype in its receiver role a DataMapping exists.

⌋()

[constr_3060] Usage of networkRepresentationProps and physicalProps
⌈ Usage of networkRepresentationProps and physicalProps shall follow the restrictions given in table 2.24 .

⌋()

	Element	
Attributes of SwDataDefProps	**SystemSignal.physicalProps**	**ISignal.networkProps additionalNativeTypeQualifier**
annotation	NA	NA
baseType	NA	D
compuMethod	D	I
dataConstr	D	M
displayFormat	D	M
implementationDataType	NA	NA
invalidValue	NA	D
swAddrMethod	NA	NA
swAlignment	NA	NA
swBitRepresentation	NA	NA
swCalibrationAccess	NA	NA
swCalprmAxisSet	NA	NA
swCalprmAxisSet. swCalprm Axis /SwAxisGrouped. swCal-prmRef	NA	NA
swCalprmAxisSet. swCalprm Axis /SwAxisIndividual. sw VariableRef	NA	NA
swCalprmAxisSet. swCalprm Axis /SwAxisGrouped. shared AxisType	NA	NA
swCalprmAxisSet. swCalprm Axis /SwAxisIndividual. input VariableType	NA	NA
swCalprmAxisSet/ AxisIndi-vidual/ Unit	NA	NA
swCalprmAxisSet/ BaseType	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

▽

△

	Element	
swRecordLayout	NA	NA
swRefreshTiming	NA	NA
swTextProps	NA	NA
swValueBlockSize	NA	NA
unit	D	M
valueAxisDataType	NA	NA

Table 2.24: Allowed SwDataDefProps Attributes for the ISignal and SystemSignal

[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_3063] Usage of `portNumber` and `dynamicallyAssigned` with value "true" is mutually exclusive ⌈ Usage of `portNumber` and `dynamicallyAssigned` with value "true" is mutually exclusive.

⌋()

[constr_3064] Usage of `serviceInstance` , `eventHandler` and `eventGroup` references ⌈ The `serviceInstance` , `eventHandler` and `eventGroup` references shall only be used to describe a service based communication over the Internet Protocol. More details are described by [TPS_SYST_01151] and [TPS_SYST_01152] .

⌋()

[constr_3065] Mapping of queued `Trigger` s to `SystemSignal` s is prohibited ⌈ A `TriggerToSignalMapping` of a `Trigger` with `swImplPolicy` set to `queued` is prohibited.

⌋()

[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 ⌈ The value of CanNmCluster . nmNidPosition shall only be set to either 0 or 1.

⌋()

[constr_3070] Allowed CanNmCluster . nmCbvPosition values ⌈ 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 ⌈ The value of UdpNmCluster . nmNidPosition shall only be set to either 0 or 1.

⌋()

[constr_3079] Allowed UdpNmCluster . nmCbvPosition values ⌈ 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 each sender needs to be represented by the same `SystemSignal` .

⌋()

[constr_3087] `DataMapping` to `PRPortPrototype` ⌈ For inter-ECU communication between `SwComponentPrototype` s which involves `PRPortPrototype` s for each `DataPrototype` there shall be one `SystemSignal` and at most two `DataMapping` s, one for each direction.

⌋()

[constr_3088] `SystemSignal` that is not part of a `SystemSignalGroup` in a complete System Description ⌈ For each `SystemSignal` that is not part of a `SystemSignalGroup` in a complete `System` with `category` `SYSTEM_DESCRIPTION` exactly one `DataMapping` per `communicationDirection` shall be defined (`PPortPrototype` , `RPortPrototype` , `PRPortPrototype`). Preference: `AbstractProvidedPortPrototype`

⌋()

[constr_3089] `SystemSignal` that is part of exactly one `SystemSignalGroup` and is not transmitted additionally as standalone `SystemSignal` in a complete System Description ⌈ For each `SystemSignal` that is part of exactly one `SystemSignalGroup` and is not transmitted additionally as standalone `SystemSignal` in a complete `System` with `category` `SYSTEM_DESCRIPTION` exactly one `DataMapping` per `communicationDirection` shall be defined (`PPortPrototype` , `RPortPrototype` , `PRPortPrototype`). Preference: `AbstractProvidedPortPrototype`

⌋()

[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_3126] headerLength shall be less or equal output buffer size ⌈ The headerLength shall be less or equal of the worst case output buffer size which is specified in bufferComputation in BufferProperties .

⌋()

[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 [13] .

⌋()

[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 connectionNegotiation-

Behavior 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 . containerIPduHeaderByteOrder ⌈ The values of System . containerIPduHeaderByteOrder 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 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 with ContainerIPdu . headerType = longHeader the IPdu . containedIPduProps . headerIdLongHeader shall be defined.

⌋()

[constr_3143] Mandatory headerIdShortHeader for shortHeader ⌈ For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu . con-

tainedPduTriggering with ContainerIPdu . headerType = shortHeader the IPdu . 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) < nmNetwork-Timeout
- FlexRay: (pnResetTime + pncPrepareSleepTimer) < nmReadySleep-Time

⌋()

[constr_3148] executeDespiteDataUnavailability setting in case an E2E Transformer is used ⌈ A transformer chain using E2E shall be configured with Data-Transformation . 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_11: BufferProperties . headerLength = 16 bits

8. PROFILE_22: `BufferProperties` . `headerLength` = 16 bits

⌋()

[constr_3152] `BufferProperties` . `headerLength` settings for an E2E transformer used in combination with a COM Based transformer ⌈ An E2E 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_1: if `dataIdMode` == `lower12Bit` : 16 bits

PROFILE_1: if `dataIdMode` != `lower12Bit` : 12 bits

PROFILE_2: 16 bits

PROFILE_4: 96 bits

PROFILE_5: 24 bits

PROFILE_6: 40 bits

PROFILE_7: 160 bits

PROFILE_11: if `dataIdMode` == `lower12Bit` : 16 bits

PROFILE_11: if `dataIdMode` == `all16Bit` : 12 bits

PROFILE_22: 16 bits

⌋()

[constr_3154] `BufferProperties` . `bufferComputation` setting for an E2E transformer when used together with a Com-based transformer ⌈ The `BufferProperties` . `bufferComputation` of an E2E transformer used in a transformer chain with a COM Based transformer shall be configured in the following way:

```
<BUFFER-COMPUTATION>
  <COMPU-RATIONAL-COEFFS>
    <COMPU-NUMERATOR>
       <V>0</V>
       <V>1</V>
    </COMPU-NUMERATOR>
    <COMPU-DENOMINATOR>
       <V>1</V>
    </COMPU-DENOMINATOR>
  </COMPU-RATIONAL-COEFFS>
</BUFFER-COMPUTATION>
```

⌋()

[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 ⌈ If the EndToEndTransformationDescription . profileName attribute has a value of PROFILE_04 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 mul-

tiplicity 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 ⌈ If the EndToEndTransformationDescription . profileName attribute has a value of PROFILE_04, PROFILE_06, or PROFILE_07 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 ⌈ If the EndToEndTransformationDescription . profileName attribute has a value of PROFILE_04, PROFILE_06, or PROFILE_07 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 Com BasedTransformer 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 ⌈ If the EndToEndTransformationDescription . profileName attribute has a value of PROFILE_04, PROFILE_05, PROFILE_06, or PROFILE_07 the value of the EndToEndTransformationDescription . offset attribute shall be equal to the value of the EndToEndTransformationDescription . upperHeaderBitsToShift attribute.

⌋()

[constr_3169] Attribute multiplicities and values in PROFILE_02 and PROFILE_22 ⌈ If the EndToEndTransformationDescription . profileName attribute has a value of PROFILE_02 or PROFILE_22 then:

1. the multiplicity of the EndToEndTransformationDescription . crcOffset attribute shall be 0.
2. the multiplicity of the EndToEndTransformationDescription . counterOffset attribute shall be 0.
3. the multiplicity of the EndToEndTransformationDescription . dataIdNibbleOffset attribute shall be 0.
4. 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_11, PROFILE_22 ⌈ If the `EndToEndTransformationDescription.profileName` attribute has a value of PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, PROFILE_11, or PROFILE_22 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_3176] Value range of `windowSize` ⌈ The value of the `windowSize` attribute shall be greater or equal to 1.

⌋()

[constr_3177] Dependency between `maxErrorStateValid`, `maxErrorStateInit` and `maxErrorStateInvalid` ⌈ The following restriction shall be respected:
`maxErrorStateValid` >= `maxErrorStateInit` >= `maxErrorStateInvalid` >= 0

⌋()

[constr_3178] Dependency between `minOkStateValid`, `minOkStateInit` and `minOkStateInvalid` ⌈ The following restriction shall be respected:
1 <= `minOkStateValid` <= `minOkStateInit` <= `minOkStateInvalid`

⌋()

[constr_3179] Dependency between minOkStateInit , maxErrorStateInit and windowSize ⌈ The following restriction shall be respected:
minOkStateInit + maxErrorStateInit <= windowSize

⌋()

[constr_3180] Dependency between minOkStateValid , maxErrorStateValid and windowSize ⌈ The following restriction shall be respected:
minOkStateValid + maxErrorStateValid <= windowSize

⌋()

[constr_3181] Dependency between minOkStateInvalid , maxErrorStateInvalid and windowSize ⌈ The following restriction shall be respected: minOkStateInvalid + maxErrorStateInvalid <= windowSize

⌋()

[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_22 ⌈ If the EndToEndTransformationDescription . profileName attribute is set to a value of PROFILE_02, PROFILE_04, PRO-

FILE_05, PROFILE_06, PROFILE_07, or PROFILE_22 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_22 ⌈ If the EndToEndTransformationDescription . profileName attribute is set to a value of PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, or PROFILE_22 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_22 ⌈ If the EndToEndTransformationDescription . profileName attribute is set to a value of PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, or PROFILE_22 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_22 or dataIdMode different from lower12Bit

⌈ If the EndToEndTransformationDescription . profileName attribute is set to a value of PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, or PROFILE_22 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_22 ⌈ If the EndToEndTransformationDescription . profileName attribute is set to a value PROFILE_02, PROFILE_04, PROFILE_05, PROFILE_06, PROFILE_07, or PROFILE_22 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_3201] `eventGroupIdentifier` in `ConsumedEventGroup` s that are referenced by the same `EventHandler` ⌈ In case that an `EventHandler` refers to several `ConsumedEventGroup` s all these `ConsumedEventGroup` s shall have the same `eventGroupIdentifier` .

⌋()

[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 CanFrameTriggering s where the attribute identifier contains the identical PGN (as defined in section 5.2 Protocol Data Unit in [14]) 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 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 TransformationISignalProps does not transport a client/server communication.

⌋()

[constr_3215] TransformationTechnology . version and TransformationTechnology . protocol settings for request and response of a client/server communication ⌈ TransformationTechnology . version and Transforma-

tionTechnology . 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 TransformationTechnology . 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 Fixed-size 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 restricted 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_3241] Usage of AssignFrameId . messageId ⌈ The value of AssignFrameId . messageId for the AssignFrameId that refers to a LinSlave in the role assignedController shall be equal to the messageId of the LinConfigurableFrame aggregated by LinCommunicationConnector in role linConfigurableFrame that points to this LinSlave in the role commController .

⌋()

[constr_3242] Usage of UnassignFrameId . messageId ⌈ The value of UnassignFrameId . messageId for the UnassignFrameId that refers to a LinSlave in the role assignedController shall be equal to the messageId of the LinConfigurableFrame aggregated by LinCommunicationConnector in role linConfigurableFrame that points to this LinSlave in the role commController .

⌋()

[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 Communi-

cationConnector 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_3259] Allowed use of SdServerConfig . capabilityRecord ⌈ A TagWithOptionalValue element may only be composed (in role capabilityRecord) by a SdServerConfig element if the respective SdServerConfig element is directly composed by a ProvidedServiceInstance element in role sdServerConfig . A TagWithOptionalValue element must not be composed (in role capabil-

ityRecord) by an SdServerConfig element if the respective SdServerConfig element is composed by an EventHandler element in role sdServerConfig .

⌋()

[constr_3260] Allowed use of SdClientConfig . capabilityRecord ⌈ A TagWithOptionalValue element may only be composed (in role capabilityRecord) by a SdClientConfig element if the respective SdClientConfig element is directly composed by a ConsumedServiceInstance element in role sdClientConfig . A TagWithOptionalValue element must not be composed (in role capabilityRecord) by an SdClientConfig element if the respective SdClientConfig element is composed by a ConsumedEventGroup element in role sdClientConfig .

⌋()

[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 `SocketConnectionBundle` s ⌈ SD `SocketConnectionBundle` s defined in [TPS_SYST_02119] shall only refer to `PduTriggering` s which point to `GeneralPurposePdu` s of category SD.

⌋()

[constr_3268] Service Discovery `SocketConnectionBundle` `serverPort` reference to a `TpPort` ⌈ Each SD `SocketConnectionBundle` defined in [TPS_SYST_02119] shall refer with the `serverPort` reference to an `ApplicationEndpoint` (via `SocketAddress`) with a Udp Port.

⌋()

[constr_3269] Service Discovery `SocketConnection` `clientPort` reference to a `TpPort` ⌈ Each SD `SocketConnection` defined in [TPS_SYST_02119] shall refer with the `clientPort` reference to an `ApplicationEndpoint` (via `SocketAddress`) with Udp Port `dynamicallyAssigned` set to true.

⌋()

[constr_3270] Service Discovery `SocketConnection` `clientPort` reference to an IP Address ⌈ Each SD `SocketConnection` defined in [TPS_SYST_02119] shall refer with the `clientPort` reference to a `NetworkEndpoint` (via `SocketAddress` . `applicationEndpoint`) with IP Address ANY (IPv4 or IPv6).

⌋()

[constr_3271] `clientIpAddrFromConnectionRequest` and `clientPortFromConnectionRequest` settings for SD `SocketConnection` s ⌈ SD `SocketConnection` s defined in [TPS_SYST_02119] shall define `clientIpAddrFromConnectionRequest` set to true and `clientPortFromConnectionRequest` set to true.

⌋()

[constr_3272] `SocketConnectionIpduIdentifier` . `headerId` setting for SD `SocketConnectionBundle` s ⌈ The `SocketConnectionIpduIdentifier` . `headerId` of SD `SocketConnectionBundle` s defined in [TPS_SYST_02119] shall always be set to 0xFFFF8100 for SD messages.

⌋()

[constr_3273] Service Discovery multicast SocketConnectionBundle 's serverPort reference to an IP Address ⌈ The SD SocketConnectionBundle for multicast defined in [TPS_SYST_02119] (SocketConnectionBundle B) shall refer via the serverPort to a SocketAddress representing a Multicast Address.

⌋()

[constr_3274] Service Discovery unicast SocketConnectionBundle 's serverPort reference to an IP Address ⌈ The SD SocketConnectionBundle for unicast defined in [TPS_SYST_02119] (SocketConnectionBundle A) shall refer via the serverPort to a SocketAddress representing a Unicast Address.

⌋()

[constr_3275] PduTriggering containment in different PdurIPduGroup s of the same EcuInstance is not allowed ⌈ A PduTriggering shall not be referenced by more than one PdurIPduGroup in the role iPdu where each of these PdurIPduGroup s are referenced by the same EcuInstance .

⌋()

[constr_3276] Prohibition of usage of allowedIPv6ExtHeaders in IPv4 SocketConnection s ⌈ IPv4 SocketConnection s shall not define allowedIPv6ExtHeaders . An IPv4 SocketConnection points to a SocketAddress in the role clientPort and relates to an ApplicationEndpoint that refers to a NetworkEndpoint that has an Ipv4Configuration as networkEndpointAddress .

⌋()

[constr_3277] Restriction of usage of IPv6ExtHeaderFilterList s in IPv6 SocketConnection s ⌈ All SocketConnection s 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 SOMEIPTransformationProps 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 static size 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_3286] ISignal . length shall be consistent to transformer configuration ⌈ For ISignal s that are used for transformed data, the value ISignal . length shall be greater or equal to the maximum possible size of the transformed data (including alignment). This size can be calculated by using the formulas specified in the TransformationTechnology . bufferProperties . bufferComputation of all TransformationTechnologies in the ordered list DataTransformation .

transformerChain for the length that is determined from the mapped VariableDataPrototype .

⌋()

[constr_3297] Prohibition of usage of allowedTcpOptions in Udp SocketConnection s ⌈ Udp SocketConnection s shall not define allowedTcpOptions . A Udp SocketConnection points to a SocketAddress in the role clientPort and relates to 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] SocketConnectionBundle . pathMtuDiscoveryEnabled setting dependency ⌈ SocketConnectionBundle . pathMtuDiscoveryEnabled shall only be set to TRUE if EthernetCommunicationConnector . pathMtuEnabled == TRUE.

⌋()

[constr_3311] Usage of SocketConnectionBundle . flowLabel ⌈ SocketConnectionBundle . flowLabel shall only be used if the SocketConnectionBundle points to a SocketAddress in the role serverPort with an ApplicationEndpoint that 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_3314] BufferProperties . bufferComputation is mandatory ⌈ The BufferProperties that is aggregated by TransformationTechnology in the role bufferProperties shall always define the bufferComputation .

⌋()

[constr_3315] The value of V0 in BufferProperties . bufferComputation setting for a COM Based transformer ⌈ The value of V0 of bufferComputation of a TransformationTechnology which has the protocol attribute set to COMBased shall have the same value as the length attribute of the ISignalIPdu to which the ISignalGroup is mapped. The ISignalGroup refers to the DataTransformation in the role comBasedSignalGroupTransformation which refers to a TransformationTechnology in the transformerChain .

⌋()

[constr_3316] Allowed values for EndToEndTransformationDescription . maxDeltaCounter in PROFILE_07 ⌈ If the EndToEndTransformationDescription . profileName attribute has a value of PROFILE_07 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 ImplementationDataType s ⌈ 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 SocketConnectionIpduIdentifier . pduCollectionSemantics in the context of one SocketConnectionBundle ⌈ The value of the attribute SocketConnectionIpduIdentifier . pduCollectionSemantics shall be identical for all referenced SocketConnectionIpduIdentifier s within the context of a given SocketConnectionBundle .

⌋()

[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 and SecureCommunicationAuthenticationProps attribute values for predefined categories ⌈ Ta-

ble 2.25 defines applicable attribute values for security profiles that are standardized by AUTOSAR.

⌋()

	PROFILE		
Attributes	**PROFILE_01**	**PROFILE_02**	**PROFILE_03** `authAlgorithm`
`authInfoTxLength`	24 bits	24 bits	28 bits
`freshnessValueLength`	Not Specified	0 bits	64 bits
`freshnessValueTxLength`	8 bits	0 bits	4 bits

Table 2.25: Security Profiles that are standardized by AUTOSAR

[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 `transportPdu` shall reference a `GeneralPurposeIPdu` with category SOMEIP_SEGMENTED_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

⌋()

[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

⌋()

[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), ProvidedServiceInstance / ConsumedServiceInstance may only be defined in the ApplicationEndpoint aggregated by one of these SocketAddress entries.

⌋()

[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 meta-class 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_3400] Usage of SdClientConfig attributes in ConsumedServiceInstance and ConsumedEventGroup ⌈ Usage of SdClientConfig attributes in ConsumedServiceInstance and ConsumedEventGroup shall follow the restrictions given in 2.26 .

⌋()

	Element	
Attributes	**ConsumedServiceInstance**	**ConsumedEventGroup** `clientServiceMinorVersion`
`clientServiceMajorVersion`	1	0
`initialFindBehavior.initialDelayMinValue`	1	0
`initialFindBehavior.initialDelayMaxValue`	1	0
`initialFindBehavior.initialRepetitionsBaseDelay`	0..1	0
`initialFindBehavior.initialRepetitionsMax`	1	0
`requestResponseDelay.maxValue`	0	1
`requestResponseDelay.minValue`	0	1
`ttl`	1	1
`capabilityRecord.key`	0..1	0
`capabilityRecord.value`	0..1	0

Table 2.26: Allowed usage of `SdClientConfig` attributes in `ConsumedServiceInstance` and `ConsumedEventGroup`

[constr_3401] Usage of `SdServerConfig` attributes in `ProvidedServiceInstance` and `EventHandler` ⌈ Usage of `SdServerConfig` attributes in `ProvidedServiceInstance` and `EventHandler` shall follow the restrictions given in 2.27 .

⌋()

	Element	
Attributes	**ProvidedServiceInstance**	**EventHandler** `serverServiceMinorVersion`
`serverServiceMajorVersion`	1	0
`initialOfferBehavior.initialDelayMinValue`	1	0
`initialOfferBehavior.initialDelayMaxValue`	1	0
`initialOfferBehavior.initialRepetitionsBaseDelay`	0..1	0
`initialOfferBehavior.initialRepetitionsMax`	1	0
`offerCyclicDelay`	0..1	0
`requestResponseDelay.maxValue`	1	1
`requestResponseDelay.minValue`	1	1
`ttl`	1	1
`capabilityRecord.key`	0..1	0
`capabilityRecord.value`	0..1	0

Table 2.27: Allowed usage of `SdServerConfig` attributes in `ProvidedServiceInstance` and `EventHandler`

[constr_3402] Mandatory `offset` if `noHeader` is used ⌈ For each `IPdu` which is assigned to a `ContainerIPdu` in the role `containedPduTriggering` with `Con-`

tainerIPdu . headerType = noHeader the IPdu . 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 containedIPduProps 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_3453] Uniqueness of header Id in case of `acceptAll ContainerIPdu` ⌈ In case several `PduTriggering` s are referenced in the role `ContainerIPdu` . `containedPduTriggering` by `ContainerIPdu` s with `ContainerIPdu` . `rxAcceptContainedIPdu` = `RxAcceptContainedIPduEnum` . `acceptAll` and the `PduTriggering` s refer to

- either the same `IPdu` (with `IPdu` . `containedIPduProps`)
- or different `IPdu` s (with `IPdu` . `containedIPduProps`) and equal header Ids,

only one `PduTriggering` shall have a reference to an `IPduPort` with direction IN which is applicable for this `EcuInstance` .

⌋()

[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`) 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`) 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] DRAFT ⌈ `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_3459] Applicable `transferProperty` for group signal ⌈ If the `ISignalToIPduMapping` refers to an `ISignal` in the role `iSignal` and this `ISignal` is referenced by an `ISignalGroup` in the role `iSignal` then the `ISignalToIPduMapping` of the `ISignal` shall either

- have `transferProperty` `pending` or `triggeredOnChange` defined, or
- have no `transferProperty` defined.

⌋()

[constr_3460] Full definition of `transferProperty` for group signal ⌈ If at least one of the `ISignal` s belonging to an `ISignalGroup` has a `transferProperty` defined (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_3483] The same PhysicalChannel is not allowed to be the source and the target of managedPhysicalChannel references ⌈ If a PhysicalChannel is source of any managedPhysicalChannel reference, then it shall not be the target of any managedPhysicalChannel relation.

⌋()

[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 ⌈ The elements of a composite data type shall be mapped to single

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 [13] 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 Pdu . length for CAN L-PDUs ⌈ The Pdu . length of CAN PDUs shall be restricted to 0..8 for classic CAN L-PDUs and 0..8, 12, 16, 20, 24, 32, 48, 64 for CAN FD 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` .

⌋()

2.35 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 must 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 must be the same as the `AutosarOperationArgumentInstance` , meaning that they must 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] Specifying stimulus and response in `TimingDescriptionEventChain` ⌈ The references between `TimingDescriptionEventChain` and `TimingDescriptionEvent` playing the role `stimulus` and `response` shall not reference the same `TimingDescriptionEvent` .

⌋()

[constr_4516] Specifying event chain segments ⌈ If a `TimingDescriptionEventChain` consists of further event chain segments then at least one sequence of event chain segments shall exists from the event chain's `stimulus` to the `response` .

⌋()

[constr_4517] Referencing no further event chain segments ⌈ If a TimingDescriptionEventChain is not subdivided in further event chain segments, then the reference playing the role of segment shall reference this TimingDescriptionEventChain . In other words, an event chain without any event chain segment 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 `EOCExecutableEntityRefGroup` s, 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 `ExecutableEntity` s, referenced by all `EOCExecutableEntityRef` s respectively all `EOCEventRef` s, 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 `EOCExecutableEntityRef` s, respectively all `EOCEventRef` s, or `EOCExecutableEntityRefGroup` s 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: $\texttt{patternLength} + \texttt{patternJitter} < \texttt{patternPeriod}$.

⌋()

[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 refering 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 must be the same as the `AutosarVariableInstance` , meaning that they must 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 1) a timing description event playing the role stimulus and the value of the `category` of this referenced `TimingDescriptionEvent` is set to `LET_RELEASE` ; and 2) 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] `category` of `TimingDescriptionEvent` 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 `TimingDescriptionEvent` .

⌋()

[constr_4560] `category` of `TimingDescriptionEventChain` 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 `TimingDescriptionEventChain` .

⌋()

2.36 TR_FrancaIntegration

[TR_FRANCA_CONSTR_00010] Franca connector has no duplicate links ⌈ There must not be two links with the same AUTOSAR and Franca sides in a Franca connector. ⌋()

[TR_FRANCA_CONSTR_00020] Franca connector has no client server fan out ⌈
A required client server port of an AUTOSAR component prototype must not be connected to more than one Franca instance. ⌋*()*

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, *AutosarDataType*, *BaseType*, BlueprintMappingSet, BswEntryRelationshipSet, BswModuleDescription, BswModuleEntry, BuildActionManifest, CalibrationParameterValueSet, ClientIdDefinitionSet, ClientServerInterfaceToBswModuleEntryBlueprintMapping, Collection, CompuMethod, ConsistencyNeedsBlueprintSet, ConstantSpecification, ConstantSpecificationMappingSet, CryptoServiceCertificate, CryptoServiceKey, CryptoServicePrimitive, DataConstr, DataExchangePoint, DataTransformationSet, DataTypeMappingSet, *DiagnosticCommonElement*, DiagnosticConnection, DiagnosticContributionSet, DiagnosticMasterToSlaveEventMappingSet, Documentation, EcucDefinitionCollection, EcucDestinationUriDefSet, EcucModuleConfigurationValues, EcucModuleDef, EcucValueCollection, EndToEndProtectionSet, EvaluatedVariantSet, FMFeature, FMFeatureMap, FMFeatureModel, FMFeatureSelectionSet, FlatMap, GeneralPurposeConnection, HwCategory, HwElement, HwType, IPv6ExtHeaderFilterSet, *Implementation*, InterpolationRoutineMappingSet, J1939ControllerApplication, KeywordSet, LifeCycleInfoSet, LifeCycleStateDefinitionGroup, McFunction, McGroup, ModeDeclarationGroup, ModeDeclarationMappingSet, PhysicalDimension, PhysicalDimensionMappingSet, *PortInterface*, PortInterfaceMappingSet, PortPrototypeBlueprint, PostBuildVariantCriterion, PostBuildVariantCriterionValueSet, PredefinedVariant, RapidPrototypingScenario, SdgDef, SwAddrMethod, SwAxisType, *SwComponentType*, SwRecordLayout, SwSystemconst, SwSystemconstantValueSet, SwcBswMapping, System, SystemSignal, SystemSignalGroup, TcpOptionFilterSet, *TimingExtension*, TransformationPropsSet, Unit, UnitGroup, ViewMapSet			
Attribute	***Type***	***Mul.***	***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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
arPackage	ARPackage	*	aggr	This represents a sub package within an ARPackage, thus allowing for an unlimited package hierarchy. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=blueprintDerivationTime xml.sequenceOffset=30
element	PackageableElement	*	aggr	Elements that are part of this package **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=20

▽

△

Class	**ARPackage**			
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=shortLabel xml.sequenceOffset=10

Table A.2: ARPackage

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			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.3: AbstractAccessPoint

Class	**AbstractClassTailoring**			
Package	M2::AUTOSARTemplates::StandardizationTemplate::DataExchangePoint::DataFormatTailoring			
Note	Tailoring of abststract classes in the AUTOSAR meta-model			
Base	*ARObject*, *ClassTailoring*, *DataFormatElementReference*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *SpecElementReference*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.4: 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				
Attribute	***Type***	***Mul.***	***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.5: AbstractEnumerationValueVariationPoint

Class	***AbstractEthernetFrame*** (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication			
Note	Ethernet specific attributes to the Frame.			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Frame*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	GenericEthernetFrame, Ieee1722TpEthernetFrame, UserDefinedEthernetFrame			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.6: 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***	***Mul.***	***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.7: AbstractEvent

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***	***Mul.***	***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.
upperMultiplicity Infinite	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.8: 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*, *PortPrototype*, *Referrable*			
Subclasses	PPortPrototype, PRPortPrototype			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
providedComSpec	PPortComSpec	*	aggr	Provided communication attributes per interface element (data element or operation).

Table A.9: 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*, *PortPrototype*, *Referrable*			
Subclasses	PRPortPrototype, RPortPrototype			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
requiredComSpec	RPortComSpec	*	aggr	Required communication attributes, one for each interface element.

Table A.10: AbstractRequiredPortPrototype

Class	**AccessCount**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::AccessCount			
Note	This meta-class provides one count value for a AbstractAccessPoint.			
Base	*ARObject*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
accessPoint	AbstractAccessPoint	1	ref	AbstractAccessPoint for which the count value is applicable.
value	PositiveInteger	1	attr	This attribute represents the number of determined accesses **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table A.11: AccessCount

Class	**AdminData**
Package	M2::MSR::AsamHdo::AdminData
Note	AdminData represents the ability to express administrative information 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 four 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

▽

△

Class	**AdminData**			
Base	*ARObject*			
Attribute	***Type***	***Mul.***	***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
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. **Tags:** 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.12: 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*			
Attribute	***Type***	***Mul.***	***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.13: AgeConstraint

Class	**AggregationCondition**			
Package	M2::AUTOSARTemplates::StandardizationTemplate::DataExchangePoint::DataFormatTailoring			
Note	The AggregationCondition evaluates to true, if the referenced aggregation is accepted by all rules of this condition.			
Base	*ARObject*, *AbstractCondition*, *AbstractMultiplicityRestriction*, *AttributeCondition*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
aggregation	AggregationTailoring	1	ref	The aggregation that has to be accepted by the restrictions of this AggregationCondition

Table A.14: AggregationCondition

Class	**AggregationTailoring**			
Package	M2::AUTOSARTemplates::StandardizationTemplate::DataExchangePoint::DataFormatTailoring			
Note	Tailoring of aggregations in the AUTOSAR meta-model			
Base	*ARObject*, *AttributeTailoring*, *DataFormatElementReference*, *DataFormatElementScope*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *SpecElementReference*, *SpecElementScope*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
typeTailoring	ClassTailoring	*	aggr	Local class tailoring which is applied if the content is contained by this aggegation.

Table A.15: 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*			
Attribute	***Type***	***Mul.***	***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. **Tags:** xml.sequenceOffset=10

Table A.16: 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*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
aliasName	AliasNameAssignment	1..*	aggr	AliasNames contained in the AliasNameSet. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortLabel vh.latestBindingTime=preCompileTime

Table A.17: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
bestCase ExecutionTime	MultidimensionalTime	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	1	aggr	The worst case execution time (WCET) defines the maximum amount of time the related executable entity requires for its execution.

Table A.18: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
base	AtpClassifier	1	ref	This is the base from which navigation path begins. **Stereotypes:** atpDerived
contextElement	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.19: AnyInstanceRef

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*, *AtpBlueprintable*, *AtpClassifier*, *AtpType*, *AutosarDataType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dynamicArraySizeProfile	String	0..1	attr	Specifies the profile which the array will follow if it is a variable size array.
element	ApplicationArrayElement	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.20: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
arraySizeHandling	ArraySizeHandlingEnum	0..1	attr	The way how the size of the array is handled.
arraySizeSemantics	ArraySizeSemanticsEnum	0..1	attr	This attribute controls how the information about the array size shall be interpreted.
indexDataType	ApplicationPrimitiveDataType	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.
maxNumberOfElements	PositiveInteger	0..1	attr	The maximum number of elements that the array can contain. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table A.21: 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			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.22: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
application Composite Element	ApplicationComposite ElementDataPrototype	1	iref	This represents the referenced ApplicationComposite DataPrototype.

Table A.23: 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			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
type	ApplicationDataType	1	tref	This represents the corresponding data type. **Stereotypes:** isOfType

Table A.24: ApplicationCompositeElementDataPrototype

Class	**ApplicationCompositeElementInPortInterfaceInstanceRef**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface::InstanceRefs			
Note				
Base	*ARObject*, *AtpInstanceRef*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
base	DataInterface	1	ref	This represents the SenderReceiverInterface that acts as the base in this InstanceRef definition **Stereotypes:** atpDerived **Tags:** xml.sequenceOffset=10
contextData Prototype	ApplicationComposite ElementDataPrototype	*	ref	This represents a context ApplicationCompositeData Prototype **Tags:** xml.sequenceOffset=20
rootData Prototype	AutosarDataPrototype	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	1	ref	This represents the referenced ApplicationComposite DataPrototype. **Tags:** xml.sequenceOffset=30

Table A.25: 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			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.26: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
consumed ServiceInstance	ConsumedService Instance	*	aggr	Consumed service instances.
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.
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. **Tags:** atp.Status=draft
tpConfiguration	TransportProtocol Configuration	0..1	aggr	Configuration of the used transport protocol.

Table A.27: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
errorCode	Integer	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.28: 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*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
–	–	–	–	–

Table A.29: 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*			
Attribute	**Type**	**Mul.**	**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.30: 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*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
–	–	–	–	–

Table A.31: 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*, *AtpBlueprintable*, *AtpClassifier*, *AtpType*, *AutosarDataType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
element (ordered)	ApplicationRecordElement	1..*	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 ApplicationrecordDataType. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table A.32: 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*			
Attribute	**Type**	**Mul.**	**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 ApplicationRecordElement 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. **Tags:** atp.Status=draft

Table A.33: 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*, *ValueSpecification*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
category	Identifier	1	attr	This represents the category of the RuleBasedValue Specification **Tags:** xml.sequenceOffset=-20

▽

△

Class	**ApplicationRuleBasedValueSpecification**			
swAxis Cont (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.34: 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*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SwComponentType*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.35: 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*, *ValueSpecification*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
category	Identifier	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].
swAxis Cont (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 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	SwValueCont	0..1	aggr	This represents the values of a Compound Primitive Data Type.

Table A.36: 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 ImplementationDataType Element (intentionally) isn't derived from AtpPrototype.			
Base	*ARObject*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
contextDataPrototype (ordered)	ImplementationDataTypeElement	*	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	ImplementationDataTypeElement	1	ref	This reference points to the target ImplementationDataTypeElement.

Table A.37: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
contextDataPrototype (ordered)	ImplementationDataTypeElement	*	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
rootVariableDataPrototype	VariableDataPrototype	0..1	ref	This refers to the VariableDataPrototype typed by the ImplementationDatatype in which the target can be found. **Tags:** xml.sequenceOffset=20
targetDataPrototype	ImplementationDataTypeElement	1	ref	This reference points to the target ImplementationDataTypeElement. **Tags:** xml.sequenceOffset=40

Table A.38: ArVariableInImplementationDataInstanceRef

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
direction	ArgumentDirection Enum	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.39: ArgumentDataPrototype

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.
Literal	***Description***
fixedSize	This means that the ApplicationArrayDataType will always have a fixed number of elements. **Tags:** atp.EnumerationValue=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.EnumerationValue=1

Table A.40: ArraySizeSemanticsEnum

Class	**ArrayValueSpecification**			
Package	M2::AUTOSARTemplates::CommonStructure::Constants			
Note	Specifies the values for an array.			
Base	*ARObject*, *CompositeValueSpecification*, *ValueSpecification*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
element (ordered)	ValueSpecification	*	aggr	The value for a single array element. All Value Specifications aggregated by ArrayValueSpecification shall have the same structure. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table A.41: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
provider	AbstractProvidedPort Prototype	0..1	iref	Instance of providing port.
requester	AbstractRequiredPort Prototype	0..1	iref	Instance of requiring port.

Table A.42: AssemblySwConnector

Class	**AssignFrameId**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	Schedule entry for an Assign Frame Id master request.			
Base	*ARObject*, *LinConfigurationEntry*, *ScheduleTableEntry*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
assignedFrame Triggering	LinFrameTriggering	1	ref	The frame whose identifier is set by this assignment.
messageId	PositiveInteger	0..1	attr	MessageId of the referenced frame.

Table A.43: AssignFrameId

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.44: 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 Wait Point or can lead to the invocation of a RunnableEntity.			
Base	*ARObject*, *AbstractAccessPoint*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

▽

△

Class	**AsynchronousServerCallResultPoint**			
asynchronous ServerCallPoint	AsynchronousServer CallPoint	1	ref	The referenced Asynchronous Server Call Point defines the asynchronous server call from which the results are returned.

Table A.45: 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*, *MultilanguageReferrable*, *RTEEvent*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
eventSource	AsynchronousServer CallResultPoint	1	ref	The referenced AsynchronousServerCallResultPoint which is raises the RTEEvent in case of returning asynchronous server call.

Table A.46: 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*			
Subclasses	ApplicationSwComponentType, ComplexDeviceDriverSwComponentType, EcuAbstractionSwComponent Type, NvBlockSwComponentType, SensorActuatorSwComponentType, ServiceProxySwComponent Type, ServiceSwComponentType			
Attribute	***Type***	***Mul.***	***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, variationPoint.short Label vh.latestBindingTime=preCompileTime
symbolProps	SymbolProps	0..1	aggr	This represents the SymbolProps for the AtomicSw ComponentType. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=shortName

Table A.47: AtomicSwComponentType

Class	***AtpBlueprint*** (abstract)
Package	M2::AUTOSARTemplates::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, BswModule Entry, *BuildActionEntity*, BuildActionEnvironment, BuildActionManifest, ClientServerInterfaceToBsw ModuleEntryBlueprintMapping, CompuMethod, ConsistencyNeeds, DataConstr, DataTypeMappingSet, EcucDefinitionCollection, EcucDestinationUriDefSet, EcucModuleDef, FlatMap, KeywordSet, LifeCycle State, LifeCycleStateDefinitionGroup, ModeDeclarationGroup, *PortInterface*, *PortInterfaceMapping*, Port InterfaceMappingSet, PortPrototypeBlueprint, SwAddrMethod, SwBaseType, *SwComponentType*, Vfb Timing

▽

△

Class	***AtpBlueprint*** (abstract)			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
blueprintPolicy	BlueprintPolicy	*	aggr	This role indicates whether the blueprintable element will be modifiable or not motifiable.

Table A.48: AtpBlueprint

Class	***AtpBlueprintMapping*** (abstract)			
Package	M2::AUTOSARTemplates::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			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
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.49: AtpBlueprintMapping

Class	***AttributeTailoring*** (abstract)			
Package	M2::AUTOSARTemplates::StandardizationTemplate::DataExchangePoint::DataFormatTailoring			
Note	Tailoring of Attributes			
Base	*ARObject*, *DataFormatElementReference*, *DataFormatElementScope*, *Identifiable*, *Multilanguage Referrable*, *Referrable*, *SpecElementReference*, *SpecElementScope*			
Subclasses	AggregationTailoring, PrimitiveAttributeTailoring, ReferenceTailoring			
Attribute	***Type***	***Mul.***	***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.50: 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, Unlimited IntegerValueVariationPoint

▽

△

Class	«atpMixedString» ***AttributeValueVariationPoint*** (abstract)			
Attribute	***Type***	***Mul.***	***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.51: 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			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
type	AutosarDataType	1	tref	This represents the corresponding data type. **Stereotypes:** isOfType

Table A.52: AutosarDataPrototype

Class	***AutosarDataType*** (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::Datatypes			
Note	Abstract base class for user defined AUTOSAR data types for ECU software.			
Base	*ARElement*, *ARObject*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	*AbstractImplementationDataType*, *ApplicationDataType*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
swDataDefProps	SwDataDefProps	0..1	aggr	The properties of this AutosarDataType.

Table A.53: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
operation Argument Instance	DataPrototype	1	iref	This is the reference to the instanceRef definition.

Table A.54: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
autosar Parameter	DataPrototype	0..1	iref	This instance reference is used if the callibration parameter is either imported via a port or is part of a composite data structure.
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 ▽

▽

△

Class	AutosarParameterRef			
				△ 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.55: 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*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
variableInstance	DataPrototype	1	iref	This is the reference to the instanceRef definition.

Table A.56: 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*			
Attribute	**Type**	**Mul.**	**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.
autosarVariable InImplDatatype	ArVariableIn ImplementationData InstanceRef	0..1	aggr	This is used if the target variable is inside of variableData Prototype typed by an ImplementationDataType.

▽

△

Class	**AutosarVariableRef**			
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 expclicit reference is provided for this case.

Table A.57: AutosarVariableRef

Class	***BaseType*** (abstract)			
Package	M2::MSR::AsamHdo::BaseTypes			
Note	This abstract meta-class represents the ability to specify a platform dependant base type.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Subclasses	SwBaseType			
Attribute	***Type***	***Mul.***	***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.58: BaseType

Class	**BaseTypeDirectDefinition**			
Package	M2::MSR::AsamHdo::BaseTypes			
Note	This BaseType is defined directly (as opposite to a derived BaseType)			
Base	*ARObject*, *BaseTypeDefinition*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
baseType Encoding	BaseTypeEncoding String	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

▽

△

Class	**BaseTypeDirectDefinition**			

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"`
				`nativeDeclaration: "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.
				This is required to ensure the consistent handling and interpretation by software components, RTE, COM and MCM systems.
				Tags: xml.sequenceOffset=120

Table A.59: BaseTypeDirectDefinition

Class	**Baseline**			
Package	M2::AUTOSARTemplates::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*			
Attribute	***Type***	***Mul.***	***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 tof 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.60: Baseline

Class	**BlueprintMapping**			
Package	M2::AUTOSARTemplates::StandardizationTemplate::BlueprintDedicated::GenericBlueprint			
Note	This meta-class represents the ability to map two an object and its blueprint.			
Base	*ARObject*, *AtpBlueprintMapping*			
Attribute	***Type***	***Mul.***	***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.61: BlueprintMapping

Class	***BlueprintPolicy*** (abstract)			
Package	M2::AUTOSARTemplates::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			
Attribute	***Type***	***Mul.***	***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.62: BlueprintPolicy

Class	**BlueprintPolicyNotModifiable**			
Package	M2::AUTOSARTemplates::StandardizationTemplate::AbstractBlueprintStructure			
Note	The class represents that the related attribute is not modifiable during the blueprinting.			
Base	*ARObject*, *BlueprintPolicy*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.63: BlueprintPolicyNotModifiable

Class	**BswAsynchronousServerCallPoint**			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	Represents an asynchronous procedure call point via the BSW Scheduler.			
Base	*ARObject*, *BswModuleCallPoint*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
calledEntry	BswModuleClientServerEntry	1	ref	The entry to be called.

Table A.64: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
asynchronous ServerCallPoint	BswAsynchronous ServerCallPoint	1	ref	The call point invoking the call to which the result belongs.

Table A.65: BswAsynchronousServerCallResultPoint

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.66: BswCalledEntity

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			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
receivedData	VariableDataPrototype	1	ref	The data received over the BSW Scheduler using this policy.

Table A.67: 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.			
Base	*ARObject*, *BswModuleCallPoint*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
calledEntry	BswModuleEntry	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.68: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.69: BswDistinguishedPartition

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*			
Attribute	***Type***	***Mul.***	***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
startsOnEvent	BswModuleEntity	1	ref	The entity which is started by the event.

Table A.70: BswEvent

Class	**BswExclusiveAreaPolicy**			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	The ExclusiveArea for which the BSW Scheduler using this policy.			
Base	*ARObject*, *BswApiOptions*			
Attribute	***Type***	***Mul.***	***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	1	ref	The ExclusiveArea for which the BSW Scheduler using this policy.

Table A.71: BswExclusiveAreaPolicy

Enumeration	**BswExecutionContext**
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswInterfaces
Note	Specifies the execution context required or guaranteed for the call associated with this service.
Literal	***Description***
hook	Context of an OS "hook" routine always **Tags:** atp.EnumerationValue=0
interruptCat1	CAT1 interrupt context always **Tags:** atp.EnumerationValue=1
interruptCat2	CAT2 interrupt context always **Tags:** atp.EnumerationValue=2
task	Task context always **Tags:** atp.EnumerationValue=3
unspecified	The execution context is not specified by the API **Tags:** atp.EnumerationValue=4

Table A.72: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
trigger	Trigger	1	ref	The trigger associated with this event. The trigger is external to this module.

Table A.73: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
arRelease Version	RevisionLabelString	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.
behavior	BswInternalBehavior	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

▽

△

Class	**BswImplementation**			
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: <ModuleName>_<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.74: 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 BswModuleDescription.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *InternalBehavior*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

▽

△

Class	**BswInternalBehavior**			
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=shortName, 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=shortName, 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, variationPoint.shortLabel 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=shortName, 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=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=5
event	BswEvent	*	aggr	An event required by this module behavior. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=10
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, variation Point.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

▽

△

Class	**BswInternalBehavior**			
internal TriggeringPoint	BswInternalTriggering Point	*	aggr	An internal triggering point. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, 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, variation Point.shortPoint vh.latestBindingTime=preCompileTime
modeReceiver Policy	BswModeReceiver Policy	*	aggr	Implementation policy for the reception of mode switches. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=modeReceiverPolicy, variation Point.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, variation Point.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, variatioPoint.short Label vh.latestBindingTime=preCompileTime
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=atp.Splitkey shortName, variation Point.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=45

▽

△

Class	**BswInternalBehavior**			
receptionPolicy	BswDataReception Policy	*	aggr	Data reception policy for inter-partition and/or inter-core communication. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=receptionPolicy, variationPoint.short Label 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, variation Point.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, variation Point.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, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
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, variation Point.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, variation Point.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=shortName

Table A.75: BswInternalBehavior

Class	**BswInternalTriggeringPoint**			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	Represents the activation point for one or more BswInternalTriggerOccurredEvents.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***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.76: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
interrupt Category	BswInterruptCategory	1	attr	Category of the interrupt
interruptSource	String	1	attr	Allows a textual documentation of the intended interrupt source.

Table A.77: BswInterruptEntity

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
modeGroup	ModeDeclarationGroup Prototype	1	ref	This represents the ModeDeclarationGroupPrototype for which the error behavior of the mode manager applies.

Table A.78: BswModeManagerErrorEvent

Class	**BswModeSwitchEvent**			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	A BswEvent resulting from a mode switch.			
Base	*ARObject*, *AbstractEvent*, *BswEvent*, *BswScheduleEvent*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
activation	ModeActivationKind	1	attr	Kind of activation w.r.t. to the referred mode.
mode (ordered)	ModeDeclaration	1..2	iref	Reference to one or two Modes that initiate the Mode Switch Event.

Table A.79: 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 must be provided by this module.			
Base	*ARObject*, *AbstractEvent*, *BswEvent*, *BswScheduleEvent*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
modeGroup	ModeDeclarationGroup Prototype	1	ref	A mode group provided by this module. The acknowledgement of a switch of this group raises this event.

Table A.80: 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, BswSynchronousServerCallPoint			
Attribute	***Type***	***Mul.***	***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 BswDistinguishedPartitions.

Table A.81: 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 BswModuleEntry to which it refers (property encapsulatedEntry). **Tags:** atp.recommendedPackage=BswModuleEntrys			
Base	*ARObject*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
encapsulated Entry	BswModuleEntry	1	ref	The underlying BswModuleEntry. **Tags:** xml.sequenceOffset=5
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

▽

△

Class	**BswModuleClientServerEntry**			
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.82: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
serviceItem	ServiceNeeds	*	aggr	A single item (example: Nv block) for which the quality of a service is defined. The aggregation is marked as «atpSplitable» to allow for extension during the ECU configuration process. This association is deprecated since R4.0.3, since ServiceNeeds shall be associated with the new element BswServiceDependency within the BswInternalBehavior. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=shortName atp.Status=removed xml.sequenceOffset=20
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:** atpUriDef; atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=7

Table A.83: 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*

▽

△

Class	**BswModuleDescription**			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
bswModule Dependency	BswModuleDependency	*	aggr	Describes the dependency to another BSW module. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, 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, variation Point.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, variationPoint.short Label 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, variation Point.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 «atpSplitable». **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=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 parition 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=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=45
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. ▽

▽

△

Class	**BswModuleDescription**			
				△ **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, 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=shortName, 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=shortName, 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 parition 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=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=50
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=shortName, 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=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=30

▽

△

Class	**BswModuleDescription**			
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=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=40

Table A.84: 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			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
accessedMode Group	ModeDeclarationGroup Prototype	*	ref	A mode group which is accessed via API call by this entity. It must be a ModeDeclarationGroupPrototype required by this module or cluster. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
activationPoint	BswInternalTriggering Point	*	ref	Activation point used by the module entity to activate one or more internal triggers. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
callPoint	BswModuleCallPoint	*	aggr	A call point used in the code of this entitiy. The variablity 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:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
calledEntry	BswModuleEntry	*	ref	The entry of another (or the same) BSW module which is called by this entry (usually via C function call). This information allows to set up a model of call chains. The variablity 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. Note that this relation has been merked as obsolete, since the more powerful definition of a callPoint should be used. **Stereotypes:** atpVariation **Tags:** atp.Status=removed vh.latestBindingTime=preCompileTime
dataReceive Point	BswVariableAccess	*	aggr	The data is received via the BSW Scheduler. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

▽

△

Class	***BswModuleEntity*** (abstract)			
dataSendPoint	BswVariableAccess	*	aggr	The data is sent via the BSW Scheduler. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
implemented Entry	BswModuleEntry	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 must be a BswTrigger released (i.e. owned) by this module or cluster. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
managedMode Group	ModeDeclarationGroup Prototype	*	ref	A mode group which is managed by this entity. It must be a ModeDeclarationGroupPrototype provided by this module or cluster. **Stereotypes:** atpVariation **Tags:** 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.85: 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*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
argument (ordered)	SwServiceArg	*	aggr	An argument belonging to this BswModuleEntry. **Stereotypes:** atpVariation **Tags:** 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	1	attr	The type of call associated with this service. **Tags:** xml.sequenceOffset=25
execution Context	BswExecutionContext	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

▽

△

Class	**BswModuleEntry**			
functionPrototypeEmitter	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	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	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
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
swServiceImplPolicy	SwServiceImplPolicyEnum	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.86: BswModuleEntry

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
entry	BswModuleClientServerEntry	1	ref	The providedClientServerEntry invoked by this event.

Table A.87: BswOperationInvokedEvent

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.88: 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, BswExternalTriggerOccurredEvent, BswInternalTriggerOccurredEvent, BswModeManagerErrorEvent, BswModeSwitchEvent, BswModeSwitchedAckEvent, BswTimingEvent			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.89: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
assignedData	RoleBasedDataAssignment	*	aggr	Defines the role of an associated data object (owned by this module or cluster) in the context of the ServiceNeeds element. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
assignedEntryRole	RoleBasedBswModuleEntryAssignment	*	aggr	Defines the role of an associated BswModuleEntry in the context of the ServiceNeeds element. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=assignedEntryRole, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
ident	BswServiceDependencyIdent	0..1	aggr	This adds the ability to become referrable to BswService Dependency. **Tags:** atp.Status=shallBecomeMandatory xml.sequenceOffset=-100
serviceNeeds	ServiceNeeds	1	aggr	The associated ServiceNeeds.

Table A.90: BswServiceDependency

Class	**BswServiceDependencyIdent**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.91: BswServiceDependencyIdent

Class	**BswSynchronousServerCallPoint**			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	Represents a synchronous procedure call point via the BSW Scheduler.			
Base	*ARObject*, *BswModuleCallPoint*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
calledEntry	BswModuleClientServer Entry	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.92: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
period	TimeValue	1	attr	Requirement for the time period (in seconds) by which this event is triggered.

Table A.93: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
masteredTrigger	Trigger	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.

▽

△

Class	**BswTriggerDirectImplementation**			
task	Identifier	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.94: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
accessed Variable	VariableDataPrototype	1	ref	The data accessed via the BSW Scheduler.
context Limitation	BswDistinguished Partition	*	ref	The existence of this reference indicates that the variable is recevied resp. sent only in the context of the referred BswDistinguishedPartitions.

Table A.95: BswVariableAccess

Class	**BufferProperties**			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	Configuration of the buffer properties the transformer needs to work.			
Base	*ARObject*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
buffer Computation	CompuScale	0..1	aggr	If the transformer changes the size of the data, the CompuScale can be used to specify a rule to derive the size of the output data based on the size of the input data.
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.96: BufferProperties

Class	**BusMirrorChannel**			
Package	M2::AUTOSARTemplates::SystemTemplate::BusMirror			
Note	This element assigns a busMirrorNetworkId to the referenced channel.			
Base	*ARObject*			
Attribute	***Type***	***Mul.***	***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:** atpVariation **Tags:** vh.latestBindingTime=systemDesignTime

Table A.97: 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			
Attribute	**Type**	**Mul.**	**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 communcation channels only a single targetPduTriggering is supported. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild

Table A.98: 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*			
Attribute	**Type**	**Mul.**	**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.99: 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*			
Attribute	***Type***	***Mul.***	***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.100: 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*			
Attribute	***Type***	***Mul.***	***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.101: 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*			
Attribute	***Type***	***Mul.***	***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.102: BusMirrorChannelMappingUserDefined

Class	***BusspecificNmEcu*** (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	Busspecific NmEcu attributes.			
Base	*ARObject*			
Subclasses	CanNmEcu, FlexrayNmEcu, J1939NmEcu, UdpNmEcu			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.103: 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.
Literal	***Description***
mostSignificantByte First	Most significant byte shall come at the lowest address (also known as BigEndian or as Motorola-Format) **Tags:** atp.EnumerationValue=0
mostSignificantByte Last	Most significant byte shall come highest address (also known as LittleEndian or as Intel-Format) **Tags:** atp.EnumerationValue=1
opaque	For opaque data endianness conversion has to be configured to Opaque. See AUTOSAR COM Specification for more details. **Tags:** atp.EnumerationValue=2

Table A.104: ByteOrderEnum

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*			
Attribute	***Type***	***Mul.***	***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.

▽

△

Class	**CanControllerFdConfiguration**			
timeSeg1	PositiveInteger	1	attr	Specifies phase segment 1 in time quantas.
timeSeg2	PositiveInteger	1	attr	Specifies phase segment 2 in time quantas.
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.105: 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*			
Attribute	***Type***	***Mul.***	***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.106: CanControllerFdConfigurationRequirements

Class	**CanFrameTriggering**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanCommunication			
Note	CAN specific attributes to the FrameTriggering			
Base	*ARObject*, *FrameTriggering*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***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.
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.107: CanFrameTriggering

Class	**CanNmCluster**			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	Can specific NmCluster attributes			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *NmCluster*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
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.

▽

△

Class	**CanNmCluster**			
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 immeditate 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.108: CanNmCluster

Class	**CanPhysicalChannel**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	CAN bus specific physical channel attributes.			
Base	*ARObject*, *AbstractCanPhysicalChannel*, *Identifiable*, *MultilanguageReferrable*, *PhysicalChannel*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.109: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
addressing Format	CanTpAddressing FormatType	1	attr	Declares which communication addressing mode is supported.
canTpChannel	CanTpChannel	1	ref	Reference to the CanTpChannel on which this CanTp Connection is realized.
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.
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 wheter 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 timout 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.110: CanTpConnection

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*			
Attribute	***Type***	***Mul.***	***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.111: 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*			
Attribute	***Type***	***Mul.***	***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
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.112: Chapter

Class	**ClassContentConditional**			
Package	M2::AUTOSARTemplates::StandardizationTemplate::DataExchangePoint::DataFormatTailoring			
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*			
Attribute	***Type***	***Mul.***	***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

▽

△

Class	**ClassContentConditional**			
sdgTailoring	SdgTailoring	*	aggr	Specification of the applicable Special Data Group **Tags:** xml.sequenceOffset=40

Table A.113: ClassContentConditional

Class	***ClassTailoring*** (abstract)			
Package	M2::AUTOSARTemplates::StandardizationTemplate::DataExchangePoint::DataFormatTailoring			
Note	The ClassTailoring is an abstract class that allows the tailoring of its attributes, applicable constraints and Sdgs.			
Base	*ARObject*			
Subclasses	AbstractClassTailoring, ConcreteClassTailoring			
Attribute	***Type***	***Mul.***	***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
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.114: ClassTailoring

Class	**ClientComSpec**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	Client-specific communication attributes (RPortPrototype typed by ClientServerInterface).			
Base	*ARObject*, *RPortComSpec*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
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.115: 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*			
Attribute	***Type***	***Mul.***	***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.

▽

△

Class	**ClientIdDefinition**			
clientServer Operation	ClientServerOperation	1	iref	Reference to the ClientServerOperation that is called by the client.

Table A.116: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
lowerLimit	Limit	1	attr	This specifies the lower limit of the ClientIdRange. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
upperLimit	Limit	1	attr	This specifies the upper limit of the ClientIdRange. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild

Table A.117: ClientIdRange

Class	**ClientServerAnnotation**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::ApplicationAttributes			
Note	Annotation to a port regarding a certain Operation.			
Base	*ARObject*, *GeneralAnnotation*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
operation	ClientServerOperation	1	ref	This represents the ClientServerOperation that the Client ServerAnnotation corresponds to.

Table A.118: ClientServerAnnotation

Class	**ClientServerApplicationErrorMapping**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	This meta-class represents the ability to map ApplicationErrors onto each other.			
Base	*ARObject*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
firstApplication Error	ApplicationError	1	ref	This represents the first ApplicationError in the context of the ClientServerApplicationErrorMapping.
second ApplicationError	ApplicationError	1	ref	This represents the second ApplicationError in the context of the ClientServerApplicationErrorMapping.

Table A.119: ClientServerApplicationErrorMapping

Class	**ClientServerInterface**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	A client/server interface declares a number of operations that can be invoked on a server by a client. **Tags:** atp.recommendedPackage=PortInterfaces			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpBlueprintable*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *PortInterface*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
operation	ClientServerOperation	1..*	aggr	ClientServerOperation(s) of this ClientServerInterface. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=blueprintDerivationTime
possibleError	ApplicationError	*	aggr	Application errors that are defined as part of this interface.

Table A.120: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
errorMapping	ClientServerApplication ErrorMapping	*	aggr	Map two different ApplicationErrors defined in the context of two different ClientServerInterfaces.
operation Mapping	ClientServerOperation Mapping	1..*	aggr	Mapping of two ClientServerOperations in two different ClientServerInterfaces

Table A.121: ClientServerInterfaceMapping

Class	**ClientServerInterfaceToBswModuleEntryBlueprintMapping**			
Package	M2::AUTOSARTemplates::StandardizationTemplate::ClientServerInterfaceToBswModuleEntryMapping			
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 PortDefinedArguments matches the signature of the BswModule Entry. **Tags:** atp.recommendedPackage=BlueprintMappingSets			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Attribute	***Type***	***Mul.***	***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:** atpVariation **Tags:** 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:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table A.122: 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*			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
argument (ordered)	ArgumentDataPrototype	*	aggr	An argument of this ClientServerOperation **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=blueprintDerivationTime
possibleError	ApplicationError	*	ref	Possible errors that may by raised by the referring operation.

Table A.123: ClientServerOperation

Class	**ClientServerOperationBlueprintMapping**			
Package	M2::AUTOSARTemplates::StandardizationTemplate::ClientServerInterfaceToBswModuleEntryMapping			
Note	This class describes a specific mapping between a ClientServerOperation in a ClientServerInterface blueprint and a BswModuleEntry blueprint.			
Base	*ARObject*			
Attribute	*Type*	*Mul.*	*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.124: 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*			
Attribute	*Type*	*Mul.*	*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.

▽

△

Class	**ClientServerOperationMapping**			
firstOperation	ClientServerOperation	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	1	ref	Second to-be-mapped ClientServerOperation of a Client ServerInterface.

Table A.125: ClientServerOperationMapping

Class	**ClientServerToSignalMapping**			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	This element maps the ClientServerOperation to call- and return-SystemSignals.			
Base	*ARObject*, *DataMapping*			
Attribute	*Type*	*Mul.*	*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.
returnSignal	SystemSignal	0..1	ref	Reference to the returnSignal to which the OUT and INOUT ArgumentDataPrototypes are mapped. **Tags:** atp.Status=shallBecomeMandatory

Table A.126: ClientServerToSignalMapping

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*			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
supportBuffer Locking	SupportBufferLocking Enum	1	attr	This attribute is used to indicate the intended buffer locking behavior.

Table A.127: 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*

▽

△

Class	«atpVariation» ***CommunicationCluster*** (abstract)			
Subclasses	*AbstractCanCluster*, EthernetCluster, FlexrayCluster, LinCluster, UserDefinedCluster			
Attribute	***Type***	***Mul.***	***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 must be assigned to exactly one cluster, whereas a cluster may have one or more channels. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel 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.128: 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. **Tags:** atp.ManifestKind=MachineManifest			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	*AbstractCanCommunicationConnector*, EthernetCommunicationConnector, FlexrayCommunication Connector, LinCommunicationConnector, UserDefinedCommunicationConnector			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
commController	Communication Controller	1	ref	Reference to the communication controller. The CommunicationConnector and referenced CommunicationController must 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) must reference to the same controller.
ecuCommPort Instance	CommConnectorPort	*	aggr	An ECUs reception or send ports. atpVariation: If signals/PDUs/frames are variable, the corresponding ports must be variable, too. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild

▽

△

Class	***CommunicationConnector*** (abstract)			
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.129: 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			
Attribute	***Type***	***Mul.***	***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.130: CommunicationController

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
hardware Element	HwDescriptionEntity	*	ref	Reference from the ComplexDeviceDriverSwComponent Type to the description of the used HwElements.

Table A.131: 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.			
Base	*ARObject*, *MappingConstraint*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
clustered Component	SwComponent Prototype	1..*	iref	Reference to the components that have to be mapped together.
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. **Tags:** atp.Status=shallBecomeMandatory

Table A.132: 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 must be mapped together with any of the atomic SW components making up the B composition. Furthermore, A and B must be disjoint.			
Base	*ARObject*, *MappingConstraint*			
Attribute	***Type***	***Mul.***	***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. **Tags:** atp.Status=shallBecomeMandatory
separated Component	SwComponent Prototype	2	iref	The two components that have to be mapped to different ECUs

Table A.133: ComponentSeparation

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

▽

△

Class	**CompositionSwComponentType**			
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 SwComponentPrototypes is resolved post-build the deselected SwComponent Prototypes are still contained in the ECUs build but the instances are inactive in 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=shortName, variationPoint.shortLabel vh.latestBindingTime=postBuild
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=shortName, variationPoint.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 implementers mainly because the RTE generator is not concerned about the CompositionSwComponent Types. Rationale: if the mapping of ApplicationDataTypes on the delegated and inner ▽

▽

△

Class	CompositionSwComponentType			
				△ 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=shortLabel, variationPoint.shortLabel vh.latestBindingTime=codeGenerationTime

Table A.134: CompositionSwComponentType

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*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
compuInternalToPhys	Compu	0..1	aggr	This specifies the computation from internal values to physical values. **Tags:** xml.sequenceOffset=80
compuPhysToInternal	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.135: CompuMethod

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*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
desc	MultiLanguageOverview Paragraph	0..1	aggr	<desc> represents a general but brief description of the object in question. **Tags:** xml.sequenceOffset=30

▽

△

Class	**CompuScale**			
compuInverse Value	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
compuScale Contents	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
lowerLimit	Limit	0..1	attr	This specifies the lower limit of the scale. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=40
mask	PositiveInteger	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.136: CompuScale

Class	**CompuScales**			
Package	M2::MSR::AsamHdo::ComputationMethod			
Note	This meta-class represents the ability to stepwise express a computation method.			
Base	*ARObject*, *CompuContent*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
compu Scale (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:** atpVariation **Tags:** vh.latestBindingTime=blueprintDerivationTime ▽

▽

△

Class	**CompuScales**			
				△ xml.roleElement=true xml.roleWrapperElement=true xml.sequenceOffset=40 xml.typeElement=false xml.typeWrapperElement=false

Table A.137: CompuScales

Class	**ConcreteClassTailoring**			
Package	M2::AUTOSARTemplates::StandardizationTemplate::DataExchangePoint::DataFormatTailoring			
Note	Tailoring of concrete meta classes.			
Base	*ARObject*, *ClassTailoring*, *DataFormatElementReference*, *DataFormatElementScope*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *SpecElementReference*, *SpecElementScope*			
Attribute	***Type***	***Mul.***	***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.138: ConcreteClassTailoring

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*			
Attribute	***Type***	***Mul.***	***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.139: ConditionByFormula

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*			
Attribute	***Type***	***Mul.***	***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=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	**ConsistencyNeeds**			
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=shortName, 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=shortName, 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=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table A.140: ConsistencyNeeds

Class	**ConstantReference**			
Package	M2::AUTOSARTemplates::CommonStructure::Constants			
Note	Instead of defining this value inline, a constant is referenced.			
Base	*ARObject*, *ValueSpecification*			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
constant	ConstantSpecification	1	ref	The referenced constant.

Table A.141: 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*, *Packageable Element*, *Referrable*			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
valueSpec	ValueSpecification	1	aggr	Specification of an expression leading to a value for this constant.

Table A.142: ConstantSpecification

Class	**ConstantSpecificationMapping**			
Package	M2::AUTOSARTemplates::CommonStructure::Constants			
Note	This meta-class is used to create an association of two ConstantSpecifications. One Constant Specification 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*			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
applConstant	ConstantSpecification	1	ref	A ConstantSpecification defined in the application domain.
implConstant	ConstantSpecification	1	ref	A ConstantSpecification defined in the implementation domain.

Table A.143: ConstantSpecificationMapping

Class	**ConstantSpecificationMappingSet**			
Package	M2::AUTOSARTemplates::CommonStructure::Constants			
Note	This meta-class represents the ability to map two ConstantSpecifications to each others. One Constant Specification is supposed to be described in the application domain and the other should be described in the implementation domain. **Tags:** atp.recommendedPackage=ConstantSpecificationMappingSets			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
mapping	ConstantSpecification Mapping	1..*	aggr	ConstantSpecificationMappings owned by the Constant SpecificationMappingSet.

Table A.144: ConstantSpecificationMappingSet

Class	**ConsumedEventGroup**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	A Service may have event groups which can be consumed. A service consumer has to subscribe to the corresponding event-group. After the subscription the event consumer takes the role of a server and the event provider that of a client.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	*Type*	*Mul.*	*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.
eventGroup Identifier	PositiveInteger	0..1	attr	EventGroup ID. Shall be unique within one system to allow service discovery.
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.

△

Class	**ConsumedEventGroup**			
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.

Table A.145: ConsumedEventGroup

Class	**ConsumedServiceInstance**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Service instances that are consumed by the ECU that is connected via the ApplicationEndpoint to a CommunicationConnector.			
Base	*ARObject*, *AbstractServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
consumedEvent Group	ConsumedEventGroup	*	aggr	Selection of event-groups the consumer wants to subscribe for.
providedService Instance	ProvidedService Instance	1	ref	Reference to a providedServiceInstance to get the instanceIdentifier information from the ProvidedService Instance.
sdClientConfig	SdClientConfig	0..1	aggr	Service Discovery Client configuration.

Table A.146: ConsumedServiceInstance

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*			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
collection Semantics	ContainedIPdu CollectionSemantics Enum	1	attr	Defines whether this ContainedIPdu shall be collected using a last-is-best or queued semantics.
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. **Tags:** atp.Status=draft
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.

▽

△

Class	**ContainedIPduProps**			
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. **Tags:** atp.Status=draft

Table A.147: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
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.
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 maxium 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. **Tags:** atp.Status=draft

Table A.148: ContainerIPdu

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

▽

△

Class	**CouplingElement**			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
communication Cluster	EthernetCluster	1	ref	This relationship defines to which cluster the Coupling Element belongs.
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=shortName, 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.149: CouplingElement

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
connection Negotiation Behavior	EthernetConnection NegotiationEnum	0..1	attr	Specifies the connection negotiation of the CouplingPort. **Tags:** atp.Status=shallBecomeMandatory
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.
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.
physicalLayer Type	EthernetPhysicalLayer TypeEnum	0..1	attr	Specifies the physical layer type of the CouplingPort.

▽

△

Class	**CouplingPort**			
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.

Table A.150: CouplingPort

Class	**CouplingPortConnection**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Connection between two CouplingPorts (firstPort and secondPort).			
Base	*ARObject*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
firstPort	CouplingPort	1	ref	Reference to the first CouplingPort that is connected via the CouplingPortConnection.
secondPort	CouplingPort	1	ref	Reference to the second CouplingPort that is connected via the CouplingPortConnection.

Table A.151: 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*			
Attribute	***Type***	***Mul.***	***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.152: CouplingPortDetails

Class	**CryptoServiceCertificate**			
Package	M2::AUTOSARTemplates::SystemTemplate::SecureCommunication			
Note	This meta-class represents the ability to model a cryptographic certificate. **Tags:** atp.ManifestKind=ServiceInstanceManifest atp.Status=draft atp.recommendedPackage=CryptoServiceCertificates			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Attribute	***Type***	***Mul.***	***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.
nextHigher Certificate	CryptoService Certificate	0..1	ref	The reference identifies the next higher certificate in the certificate chain. **Tags:** atp.Status=draft

Table A.153: CryptoServiceCertificate

Class	**CryptoServiceKey**			
Package	M2::AUTOSARTemplates::SystemTemplate::SecureCommunication			
Note	This meta-class has the ability to represent a crypto key **Tags:** atp.ManifestKind=ServiceInstanceManifest atp.recommendedPackage=CryptoDevelopmentKeys			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Attribute	***Type***	***Mul.***	***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.

Table A.154: CryptoServiceKey

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*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Attribute	**Type**	**Mul.**	**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.155: 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*			
Attribute	**Type**	**Mul.**	**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.156: 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*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
dataFilterType	DataFilterTypeEnum	1	attr	This attribute specifies the type of the filter.
mask	UnlimitedInteger	0..1	attr	Mask for old and new value.
max	UnlimitedInteger	0..1	attr	Value to specify the upper boundary
min	UnlimitedInteger	0..1	attr	Value to specify the lower boundary

△

Class	**DataFilter**			
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.157: DataFilter

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			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.158: 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			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
communicationDirection	CommunicationDirectionType	0..1	attr	This attribute controls the direction into which the mapped SystemSignal is communicated with respect to the kind of PortPrototype used as the context element of the Data Mapping.
eventGroup	ConsumedEventGroup	*	ref	Via this reference a connection between the VFB View and the Ethernet EventGroups can be created.
eventHandler	EventHandler	*	ref	Via this reference a connection between the VFB View and the Ethernet EventHandlers can be created.
introduction	DocumentationBlock	0..1	aggr	This represents introductory documentation about the data mapping.
serviceInstance	AbstractServiceInstance	*	ref	Via this reference a connection between the VFB View and the Ethernet Services can be created.

Table A.159: 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*

▽

△

Class	***DataPrototype*** (abstract)			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
swDataDef Props	SwDataDefProps	0..1	aggr	This property allows to specify data definition properties which apply on data prototype level.

Table A.160: DataPrototype

Class	**DataPrototypeGroup**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::ImplicitCommunicationBehavior			
Note	This meta-class represents the ability to define a collection of DataPrototypes that are subject to the formal definition of implicit communication behavior. The definition of the collection can be nested.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dataPrototype Group	DataPrototypeGroup	*	iref	This represents the ability to define nested groups of VariableDataPrototypes. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
implicitData Access	VariableDataPrototype	*	iref	This represents a collection of VariableDataPrototypes that belong to the enclosing DataPrototypeGroup **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table A.161: DataPrototypeGroup

Class	**DataPrototypeMapping**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	Defines the mapping of two particular VariableDataPrototypes, ParameterDataPrototypes or Argument DataPrototypes with unequal names and/or unequal semantic (resolution or range) in context of two different SenderReceiverInterface, NvDataInterface or ParameterInterface or Operations. If the semantic is unequal 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
firstData Prototype	AutosarDataPrototype	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 ▽

▽

△

Class	DataPrototypeMapping			
				△ 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	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.162: DataPrototypeMapping

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*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
dataPrototype Ref	DataPrototypeInSystem Ref	0..1	aggr	Reference to a DataPrototype that is transported in the serialized ISignal.
network Representation Props	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.
transformation Props	TransformationProps	0..1	ref	Collection of AutosarDataPrototype related configuration settings for a transformer.

Table A.163: DataPrototypeTransformationProps

Class	DataReceivedEvent			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents			
Note	The event is raised when the referenced data elements are received.			
Base	*ARObject*, *AbstractEvent*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *RTEEvent*, *Referrable*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
data	VariableDataPrototype	0..1	iref	Data element referenced by event

Table A.164: DataReceivedEvent

Class	**DataSendCompletedEvent**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents			
Note	The event is raised when the referenced data elements have been sent or an error occurs.			
Base	*ARObject*, *AbstractEvent*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *RTEEvent*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
eventSource	VariableAccess	1	ref	The variable access that triggers the event.

Table A.165: 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*			
Attribute	***Type***	***Mul.***	***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.166: DataTransformation

Enumeration	**DataTransformationKindEnum**
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer
Note	This enumeration contributes to the definition of the scope of the DataTransformation.
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.EnumerationValue=0
asymmetricToByte Array	The DataTransformation shall be applied to the sending end only, i.e. from data type to byte array. **Tags:** atp.EnumerationValue=1
symmetric	The DataTransformation shall be applied at both the sending and the receiving end of the communication. **Tags:** atp.EnumerationValue=2

Table A.167: DataTransformationKindEnum

Class	**DataTypeMap**
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::Datatypes
Note	This class represents the relationship between ApplicationDataType and its implementing Abstract ImplementationDataType.
Base	*ARObject*

▽

△

Class	**DataTypeMap**			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
applicationDataType	ApplicationDataType	1	ref	This is the corresponding ApplicationDataType
implementationDataType	AbstractImplementationDataType	1	ref	This is the corresponding AbstractImplementationDataType.

Table A.168: 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*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dataTypeMap	DataTypeMap	*	aggr	This is one particular association between an ApplicationDataType and its AbstractImplementationDataType.
modeRequestTypeMap	ModeRequestTypeMap	*	aggr	This is one particular association between an ModeDeclarationGroup and its AbstractImplementationDataType.

Table A.169: DataTypeMappingSet

Class	**DataWriteCompletedEvent**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents			
Note	This event is raised if an implicit write access was successful or an error occurred.			
Base	*ARObject*, *AbstractEvent*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *RTEEvent*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
eventSource	VariableAccess	1	ref	The variable access that triggers the event.

Table A.170: 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*, *PackageableElement*, *Pdu*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
diagPduType	DiagPduType	1	attr	Attribute is used to distinguish a request from a response.

Table A.171: 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 CompositionSw ComponentType.			
Base	*ARObject*, *GeneralAnnotation*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
signalFan	SignalFanEnum	0..1	attr	Specifies the Signal Fan In or Signal Fan Out inside the Composition Type.

Table A.172: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
innerPort	PortPrototype	1	iref	The port that belongs to the ComponentPrototype in the composition **Tags:** xml.typeElement=true
outerPort	PortPrototype	1	ref	The port that is located on the outside of the Composition Type

Table A.173: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
artifact Descriptor	AutosarEngineering Object	1	aggr	The specified artifact needs to exist.
usage	DependencyUsage Enum	1..*	attr	Specification for which process step(s) this dependency is required.

Table A.174: DependencyOnArtifact

Class	***DiagEventDebounceAlgorithm*** (abstract)
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds
Note	This class represents the ability to specify the pre-debounce algorithm which is selected and/or required by the particular monitor. This class inherits from Identifiable in order to allow further documentation of the expected or implemented debouncing and to use the category for the identification of the expected / implemented debouncing.
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*

△

Class	***DiagEventDebounceAlgorithm*** (abstract)			
Subclasses	DiagEventDebounceCounterBased, DiagEventDebounceMonitorInternal, DiagEventDebounceTime Based			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.175: DiagEventDebounceAlgorithm

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 DemDebounce CounterBased.			
Base	*ARObject*, *DiagEventDebounceAlgorithm*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***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	1	attr	This value shall be taken to decrement the internal debounce counter.
counterFailed Threshold	Integer	1	attr	This value defines the event-specific limit that indicates the "failed" counter status.
counter IncrementStep Size	Integer	1	attr	This value shall be taken to increment the internal debounce counter.
counterJump Down	Boolean	1	attr	This value activates or deactivates the counter jump-down behavior.
counterJump DownValue	Integer	1	attr	This value represents the initial value of the internal debounce counter if the counting direction changes from incrementing to decrementing.
counterJumpUp	Boolean	1	attr	This value activates or deactivates the counter jump-up behavior.
counterJumpUp Value	Integer	1	attr	This value represents the initial value of the internal debounce counter if the counting direction changes from decrementing to incrementing.
counterPassed Threshold	Integer	1	attr	This value defines the event-specific limit that indicates the "passed" counter status.

Table A.176: DiagEventDebounceCounterBased

Class	**DiagEventDebounceMonitorInternal**
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds
Note	This meta-class represents the ability to indicate that the pre-debounce algorithm shall be used by the Dem for this diagnostic monitor. This is related to setting the EcuC choice container DemDebounceAlgorithmClass to DemDebounce MonitorInternal.

▽

▽

△

Class	DiagEventDebounceMonitorInternal			
	△ If the FaultDetectionAlogrithm is already known to be implemented by a specific BswModuleEntry the reference bswModuleEntry points to the function specification. If the FaultDetectionCounter value is accessible at a PortPrototype this PortPrototype shall be referenced by an assignedPort.			
Base	*ARObject*, *DiagEventDebounceAlgorithm*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
–	–	–	–	–

Table A.177: DiagEventDebounceMonitorInternal

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*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
diagnosticSession	DiagnosticSession	*	ref	This represents the associated DiagnosticSessions
environmentalCondition	DiagnosticEnvironmentalCondition	0..1	ref	This represents the environmental conditions associated with the access permission.
securityLevel	DiagnosticSecurityLevel	*	ref	This represents the associated DiagnosticSecurityLevels

Table A.178: DiagnosticAccessPermission

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*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
comControlClass	DiagnosticComControlClass	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.
customSubFunctionNumber	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.179: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
specificChannel	CommunicationCluster	1	ref	This represents the affected CommunicationClusters in the role specificChannel
subnetNumber	PositiveInteger	1	attr	This represents the applicable subnet number (which is an arbitrary number ranging from 1..14)

Table A.180: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
subNode Channel	CommunicationCluster	1	ref	This represents the affected CommunicationClusters in the role subNodeChannel
subNode Number	PositiveInteger	1	attr	This represents the applicable subNode number. The value corresponds to the request message parameter nodeIdentificationNumber of diagnostic service CommunicationControl (0x28).

Table A.181: DiagnosticComControlSubNodeChannel

Enumeration	**DiagnosticCompareTypeEnum**
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition
Note	Enumeration for the type of a comparison of values usually expressed by the following operators: ==, !=, <, <=, >, >=
Literal	***Description***
isEqual	equal **Tags:** atp.EnumerationValue=0
isGreaterOrEqual	greater than or equal **Tags:** atp.EnumerationValue=5
isGreaterThan	greater than **Tags:** atp.EnumerationValue=4
isLessOrEqual	less than or equal **Tags:** atp.EnumerationValue=3
isLessThan	less than **Tags:** atp.EnumerationValue=2
isNotEqual	not equal **Tags:** atp.EnumerationValue=1

Table A.182: DiagnosticCompareTypeEnum

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*			
Attribute	***Type***	***Mul.***	***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.

Table A.183: 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*			
Attribute	***Type***	***Mul.***	***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, 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, variationPoint.short Label vh.latestBindingTime=postBuild

Table A.184: DiagnosticContributionSet

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, DiagnosticWriteDataByIdentifier			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dataIdentifier	DiagnosticAbstractDataIdentifier	1	ref	This represents the linked DiagnosticDataIdentifier.

Table A.185: 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*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
arraySizeSemantics	ArraySizeSemanticsEnum	0..1	attr	This attribute controls the meaning of the value of the array size.
maxNumberOfElements	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 DiagnosticDataElement if used with DiagnosticReadScalingDataByIdentifier
swDataDefProps	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.

Table A.186: 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*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dataElement	DiagnosticParameter	1..*	aggr	This is the dataElement associated with the Diagnostic DataIdentifier. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=dataElement, variationPoint.shortLabel vh.latestBindingTime=postBuild
didSize	PositiveInteger	0..1	attr	This attribute indicates the size in bytes of the Diagnostic DataIdentifier.

▽

△

Class	**DiagnosticDataIdentifier**			
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.187: 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			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dataIdenti-fier (ordered)	DiagnosticDataIdentifier	*	ref	Reference to an orderd list of Data Identifiers.

Table A.188: DiagnosticDataIdentifierSet

Class	**DiagnosticDebounceAlgorithmProps**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticDebouncingAlgorithm			
Note	Defines properties for the debounce algorithm class.			
Base	*ARObject*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
debounce Algorithm	DiagEventDebounce Algorithm	1	aggr	This represents the actual debounce algorithm.
debounce Behavior	DiagnosticDebounce BehaviorEnum	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.
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

Table A.189: DiagnosticDebounceAlgorithmProps

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*			
Attribute	***Type***	***Mul.***	***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.

▽

△

Class	**DiagnosticDynamicallyDefineDataIdentifierClass**			
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.190: 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.
Literal	*Description*
clearDynamically DefineDataIdentifier	Clear the specified dynamic data identifier. **Tags:** atp.EnumerationValue=0
defineByIdentifier	The definition of dynamic data identifier shall be done via a reference to a diagnostic data identifier. **Tags:** atp.EnumerationValue=1
defineByMemory Address	The definition of dynamic data identifier shall be done via a reference to a memory address. **Tags:** atp.EnumerationValue=2

Table A.191: 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*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
dtcStatus AvailabilityMask	PositiveInteger	1	attr	This attribute contains the value of the DTC status availability mask.
ecuInstance	EcuInstance	*	ref	This represents the actual EcuInstance to which the information conatined in the DiagnosticEcuInstance contribute. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=ecuInstance
obdSupport	DiagnosticObdSupport Enum	1	attr	This attribute is used to specify the role (if applicable) in which the DiagnosticEcuInstance supports OBD.

Table A.192: 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*			
Attribute	***Type***	***Mul.***	***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	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.193: DiagnosticEcuReset

Class	**DiagnosticEnvConditionFormula**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::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*			
Attribute	***Type***	***Mul.***	***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	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.194: DiagnosticEnvConditionFormula

Class	**DiagnosticEnvDataCondition**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
compareValue	ValueSpecification	1	aggr	This attribute represents a fixed compare value taken to evaluate the compare condition.
dataElement	DiagnosticDataElement	1	ref	This reference represents the related diagnostic data element.

Table A.195: DiagnosticEnvDataCondition

Class	**DiagnosticEnvModeCondition**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition			
Note	DiagnosticEnvModeCondition are atomic condition based on the comparison of the active Mode Declaration in a ModeDeclarationGroupPrototype 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*			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
modeElement	DiagnosticEnvMode Element	1	ref	This reference represents both the ModeDeclaration GroupPrototype and the ModeDeclaration relevant for the mode comparison.

Table A.196: DiagnosticEnvModeCondition

Class	***DiagnosticEnvModeElement*** (abstract)			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition			
Note	All ModeDeclarations that are referenced in a DiagnosticEnvModeCondition must be defined as a DiagnosticEnvModeElement of this DignosticEnvironmentalCondition. This concept keeps the ARXML clean: It avoids that the DignosticEnvConditionFormula 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			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
–	–	–	–	–

Table A.197: DiagnosticEnvModeElement

Class	**DiagnosticEnvironmentalCondition**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition			
Note	The meta-class DignosticEnvironmentalCondition 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*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
formula	DiagnosticEnvCondition Formula	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.198: 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*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
clearEventBehavior	DiagnosticClearEventBehaviorEnum	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.
connectedIndicator	DiagnosticConnectedIndicator	*	aggr	Event specific description of Indicators. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=postBuild
eventClearAllowed	DiagnosticEventClearAllowedEnum	0..1	attr	This attribute defines whether the Dem has access to a "ClearEventAllowed" callback.
eventFailureCycleCounterThreshold	PositiveInteger	0..1	attr	This attribute defines the number of failure cycles for the event based fault confirmation. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
eventKind	DiagnosticEventKindEnum	1	attr	This attribute is used to distinguish between SWC and BSW events.
prestorageFreezeFrame	Boolean	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
prestoredFreezeframeStoredInNvm	Boolean	0..1	attr	If the Event uses a prestored freeze-frame (using the operations PrestoreFreezeFrame and ClearPrestoredFreezeFrame 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)
recoverableInSameOperationCycle	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.199: DiagnosticEvent

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*, *ServiceNeeds*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

▽

△

Class	**DiagnosticEventNeeds**			
considerPto Status	Boolean	0..1	attr	PTO (Power Take Off) has an impact on the respective emission-related event (OBD). This information shall be provided by SW-C description in order to consider the PTO relevance e.g. for readiness (PID $01) computation. For events with dtcKind set to 'nonEmmissionRelatedDtc' this attribute is typically false.
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.
dtcKind	DtcKindEnum	0..1	attr	This attribute indicates the kind of the diagnostic monitor according to the SWS Diagnostic Event Manger. This attribute applies for the UDS diagnostics use case.
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.
obdDtcNumber	PositiveInteger	0..1	attr	This represents a reasonable Diagnostic Trouble Code. This allows to predefine the Diagnostic Trouble Code, e.g. if the a function developer has received a particular requirement from the OEM or from a standardization body. This attribute applies for the OBD diagnostics use case.
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).
reportBehavior	ReportBehaviorEnum	0..1	attr	This switch indicates whether or not the BSW module is allowed to report the related Events before Dem_Init().
udsDtcNumber	PositiveInteger	0..1	attr	This represents a reasonable Diagnostic Trouble Code. This allows to predefine the Diagnostic Trouble Code, e.g. if the a function developer has received a particular requirement from the OEM or from a standardization body. This attribute applies for the UDS diagnostics use case.
usesMonitor Data	Boolean	0..1	attr	This attribute defines whether additional monitor data shall be added to the reporting of events.

Table A.200: DiagnosticEventNeeds

Class	**DiagnosticEventPortMapping**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticMapping			
Note	Defines to which SWC service ports with DiagnosticEventNeeds the DiagnosticEvent is mapped. **Tags:** atp.recommendedPackage=DiagnosticMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *DiagnosticSwMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
bswService Dependency	BswService DependencyIdent	0..1	ref	Reference to a BswServiceDependency that links Service Needs to BswModuleEntries.
diagnosticEvent	DiagnosticEvent	1	ref	Reference to the DiagnosticEvent that is assigned to SWC service ports with DiagnosticEventNeeds.
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.

Table A.201: DiagnosticEventPortMapping

Class	**DiagnosticEventToDebounceAlgorithmMapping**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
debounce Algorithm	DiagnosticDebounce AlgorithmProps	1	ref	Reference to a DebounceAlgorithm assigned to a DiagnosticEvent.
diagnosticEvent	DiagnosticEvent	1	ref	Reference to a DiagnosticEvent to which a Debounce Algorithm is assigned.

Table A.202: DiagnosticEventToDebounceAlgorithmMapping

Class	**DiagnosticEventToEnableConditionGroupMapping**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticMapping			
Note	Defines which EnableConditionGroup is applicable for a DiagnosticEvent. **Tags:** atp.recommendedPackage=DiagnosticMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
diagnosticEvent	DiagnosticEvent	1	ref	Reference to a DiagnosticEvent to which an Enable ConditionGroup is assigned.
enableCondition Group	DiagnosticEnable ConditionGroup	1	ref	Reference to an EnableConditionGroup assigned to a DiagnosticEvent.

Table A.203: DiagnosticEventToEnableConditionGroupMapping

Class	**DiagnosticEventToStorageConditionGroupMapping**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticMapping			
Note	Defines which StorageConditionGroup is applicable for a DiagnosticEvent. **Tags:** atp.recommendedPackage=DiagnosticMappings			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
diagnosticEvent	DiagnosticEvent	1	ref	Reference to a DiagnosticEvent to which a Storage ConditionGroup is assigned.
storage ConditionGroup	DiagnosticStorage ConditionGroup	1	ref	Reference to a StorageConditionGroup assigned to a DiagnosticEvent.

Table A.204: DiagnosticEventToStorageConditionGroupMapping

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*			
Attribute	***Type***	***Mul.***	***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	1	attr	This attribute specifies an unique identifier for an extended data record.
trigger	DiagnosticRecord TriggerEnum	1	attr	This attribute specifies the primary trigger to allocate an event memory entry.
update	Boolean	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.205: DiagnosticExtendedDataRecord

Class	**DiagnosticFimAliasEventGroupMapping**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Fim			
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*			
Attribute	***Type***	***Mul.***	***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.206: 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*			
Attribute	***Type***	***Mul.***	***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.207: DiagnosticFimAliasEventMapping

Class	**DiagnosticFimFunctionMapping**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::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			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMapping*, *DiagnosticSwMapping*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Attribute	***Type***	***Mul.***	***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).

Table A.208: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
customTrigger	String	0..1	attr	This attribute shall be taken to verbally describe the nature of the custom trigger.

▽

△

Class	**DiagnosticFreezeFrame**			
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	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.209: DiagnosticFreezeFrame

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
event	DiagnosticFimAlias Event	0..1	ref	This represents the alias event appllicable for the referencing inhibition source.
eventGroup	DiagnosticFimAlias EventGroup	0..1	ref	This represents the event group applicable for the referencing inhibition source.

Table A.210: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dataIdentifier	DiagnosticDataIdentifier	1	ref	This represents the corresponding DiagnosticData Identifier
freezeCurrent State	Boolean	1	attr	Setting this attribute to true represents the ability of the Dcm to execute a freezeCurrentState.
ioControlClass	DiagnosticIoControl Class	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.
resetToDefault	Boolean	1	attr	Setting this attribute to true represents the ability of the Dcm to execute a resetToDefault.
shortTerm Adjustment	Boolean	1	attr	Setting this attribute to true represents the ability of the Dcm to execute a shortTermAdjustment.

Table A.211: DiagnosticIOControl

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*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dataElement	DiagnosticParameter	*	aggr	This represents the data associated with the enclosing DiagnosticInfoType. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=dataElement
id	PositiveInteger	1	attr	This attribute represents the value of InfoType (see SAE J1979-DA).

Table A.212: 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*, *ServiceNeeds*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
currentValue	DiagnosticValueNeeds	0..1	ref	Reference to the DiagnosticValueNeeds indicating the access to the current value via signalBasedDiagnostics.
didNumber	PositiveInteger	0..1	attr	This represents a Data identifier for the diagnostic value. This allows to predefine the DID number if the a function developer has received a particular requirement from the OEM or from a standardization body.
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.213: DiagnosticIoControlNeeds

Class	**DiagnosticJ1939Node**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::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*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
nmNode	J1939NmNode	0..1	ref	This represents the reference to the "virtual Ecu" to which the enclosing DiagnosticJ1939Node is associated.

Table A.214: DiagnosticJ1939Node

Class	**DiagnosticMasterToSlaveEventMapping**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::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			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***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.215: DiagnosticMasterToSlaveEventMapping

Class	***DiagnosticMemoryDestination*** (abstract)			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode			
Note	This abstract meta-class represents a possible memory destination for a diagnostic event.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	DiagnosticMemoryDestinationMirror, DiagnosticMemoryDestinationPrimary, DiagnosticMemoryDestinationUserDefined			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.216: DiagnosticMemoryDestination

Class	**DiagnosticMemoryDestinationMirror**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode			
Note	This represents a mirror memory for a diagnostic event. **Tags:** atp.recommendedPackage=DiagnosticMemoryDestinations			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMemoryDestination*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.217: DiagnosticMemoryDestinationMirror

Class	**DiagnosticMemoryDestinationPrimary**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode			
Note	This represents a primary memory for a diagnostic event. **Tags:** atp.recommendedPackage=DiagnosticMemoryDestinations			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMemoryDestination*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.218: DiagnosticMemoryDestinationPrimary

Class	**DiagnosticMemoryDestinationUserDefined**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode			
Note	This represents a user-defined memory for a diagnostic event. **Tags:** atp.recommendedPackage=DiagnosticMemoryDestinations			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticMemoryDestination*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
memoryId	PositiveInteger	1	attr	This represents the identifier of the user-defined memory.

Table A.219: 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			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
access Permission	DiagnosticAccessPermission	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.
memoryHighAddress	PositiveInteger	0..1	attr	This represents the upper bound for addresses of the memory segment.
memoryHighAddressLabel	String	0..1	attr	This represents a symbolic label for the upper bound for addresses of the memory segment.
memoryLowAddress	PositiveInteger	0..1	attr	This represents the lower bound for addresses of the memory segment.
memoryLowAddressLabel	String	0..1	attr	This represents a symbolic label for the lower bound for addresses of the memory segment.

Table A.220: DiagnosticMemoryIdentifier

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
bitOffset	PositiveInteger	1	attr	This represents the bitOffset of the DiagnosticParameter
dataElement	DiagnosticDataElement	1	aggr	This represents the related dataElement of the Diagnostic Parameter **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=postBuild
supportInfo	DiagnosticParameterSupportInfo	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.221: DiagnosticParameter

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dataElement	DiagnosticParameter	*	aggr	This represents the data carried by the Diagnostic ParameterIdentifier. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=dataElement, variationPoint.short Label vh.latestBindingTime=postBuild
id	PositiveInteger	1	attr	This is the numerical identifier used to identify the DiagnosticParameterIdentifier in the scope of diagnostic workflow (see SAE J1979-DA).
pidSize	PositiveInteger	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	DiagnosticSupportInfo Byte	0..1	aggr	This represents the supported information associated with the DiagnosticParameterIdentifier.

Table A.222: DiagnosticParameterIdentifier

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*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
diagnostic Connection	DiagnosticConnection	*	ref	This represents the collection of applicable Diagnostic Connections for this DiagnosticProtocol. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=diagnosticConnection, variation Point.shortLabel vh.latestBindingTime=postBuild
priority	PositiveInteger	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
protocolKind	NameToken	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").

△

Class	**DiagnosticProtocol**			
serviceTable	DiagnosticServiceTable	0..1	ref	This represents the service table applicable for the given diagnostic protocol. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=serviceTable, variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.223: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
readClass	DiagnosticReadDataByIdentifierClass	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 DiagnosticReadDataByIdentifier in the given context.

Table A.224: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
maxPeriodicDidToRead	PositiveInteger	1	attr	This represents the maximum number of data identifiers that can be included in one request.
periodicRate	DiagnosticPeriodicRate	1..*	aggr	This represents the description of a collection of periodic rates in which the service can be executed.
schedulerMaxNumber	PositiveInteger	1	attr	This represents the maximum number of periodic data identifiers that can be scheduled in parallel.

Table A.225: 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*

▽

△

Class	**DiagnosticReadScalingDataByIdentifier**			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
readScaling DataClass	DiagnosticReadScaling DataByIdentifierClass	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 DiagnosticReadScalingData ByIdentifier in the given context.

Table A.226: DiagnosticReadScalingDataByIdentifier

Enumeration	**DiagnosticRecordTriggerEnum**
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticFreezeFrame
Note	Triggers to allocate an event memory entry.
Literal	***Description***
confirmed	capture on "Confirmed" **Tags:** atp.EnumerationValue=0
custom	implement custom capture **Tags:** atp.EnumerationValue=4
fdcThreshold	capture on "FDC Threshold" **Tags:** atp.EnumerationValue=1
pending	capture on "Pending" **Tags:** atp.EnumerationValue=2
testFailed	capture on "Test Failed" **Tags:** atp.EnumerationValue=3

Table A.227: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
requestControl OfOnBoard DeviceClass	DiagnosticRequest ControlOfOnBoard DeviceClass	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	1	ref	This represents the test Id for the mode 0x08.

Table A.228: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
pid	DiagnosticParameter Identifier	1	ref	This represents the PID associated with this instance of the OBD mode 0x01 service.
requestCurrent Powertrain DiagnosticData Class	DiagnosticRequest CurrentPowertrainData Class	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.229: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
request DownloadClass	DiagnosticRequest DownloadClass	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.230: DiagnosticRequestDownload

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
request	DiagnosticParameter	*	aggr	This represents the request parameters.
response	DiagnosticParameter	*	aggr	This represents the response parameters.

Table A.231: 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*, *DiagnosticMemoryAddressableRangeAccess*, *DiagnosticMemoryByAddress*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
requestUpload Class	DiagnosticRequest UploadClass	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.232: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
infoType	DiagnosticInfoType	1	ref	This represents the info type associated with the mode 0x09 service.
requestVehicle Information Class	DiagnosticRequest VehicleInfoClass	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.233: DiagnosticRequestVehicleInfo

Class	**DiagnosticResponseOnEvent**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent			
Note	This represents an instance of the "Response on Event" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticResponseOnEvents			
Base	*ARElement*, *ARObject*, *CollectableElement*, *DiagnosticCommonElement*, *DiagnosticServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
event	DiagnosticResponseOn EventTrigger	*	aggr	This represents the collection of DiagnosticResponseOn EventTriggers defined in the context of the enclosing DiagnosticResponseOnEvent.
eventWindow	DiagnosticEventWindow	*	aggr	This represents the applicable DiagnosticEventWindows
responseOn EventAction	DiagnosticResponseOn EventActionEnum	0..1	attr	Defines sub-functions of the service ResponseOnEvent.

▽

△

Class	**DiagnosticResponseOnEvent**			
responseOn EventClass	DiagnosticResponseOn EventClass	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 DiagnosticResponseOnEvent in the given context.
storeEvent Support	DiagnosticStoreEvent SupportEnum	0..1	attr	Defines how a specific event shall be handled.

Table A.234: DiagnosticResponseOnEvent

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
id	PositiveInteger	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.
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.235: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
routine	DiagnosticRoutine	1	ref	This refers to the applicable DiagnosticRoutine.

▽

△

Class	**DiagnosticRoutineControl**			
routineControl Class	DiagnosticRoutine ControlClass	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.236: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
diagRoutine Type	DiagnosticRoutineType Enum	1	attr	This denotes the type of diagnostic routine which is implemented by the referenced server port.
ridNumber	PositiveInteger	0..1	attr	This represents a routine identifier for the diagnostic routine. This allows to predefine the RID number if the a function developer has received a particular requirement from the OEM or from a standardization body.

Table A.237: DiagnosticRoutineNeeds

Enumeration	**DiagnosticRoutineTypeEnum**
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds
Note	This enumerator specifies the different types of diagnostic routines.
Literal	***Description***
asynchronous	This indicates that the diagnostic server is not blocked while the diagnostic routine is running. **Tags:** atp.EnumerationValue=0
synchronous	This indicates that the diagnostic routine blocks the diagnostic server in the ECU while the routine is running. **Tags:** atp.EnumerationValue=1

Table A.238: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
requestSeedId	PositiveInteger	1	attr	This would be 0x01, 0x03, 0x05, ... The sendKey id can be computed by adding 1 to the requestSeedId

▽

△

Class	**DiagnosticSecurityAccess**			
securityAccess Class	DiagnosticSecurity AccessClass	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.
securityLevel	DiagnosticSecurityLevel	1	ref	This reference identifies the applicable security level for the security access. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=securityLevel

Table A.239: DiagnosticSecurityAccess

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*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
accessData RecordSize	PositiveInteger	0..1	attr	This represents the size of the AccessDataRecord used in GetSeed. Unit:byte.
keySize	PositiveInteger	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	1	attr	This represents the delay time after a failed security access. Unit: second.
seedSize	PositiveInteger	1	attr	This represents the size of the security seed. Unit: byte.

Table A.240: 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*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*
Subclasses	DiagnosticClearDiagnosticInformationClass, DiagnosticClearResetEmissionRelatedInfoClass, DiagnosticComControlClass, DiagnosticControlDTCSettingClass, DiagnosticCustomServiceClass, DiagnosticDataTransferClass, DiagnosticDynamicallyDefineDataIdentifierClass, DiagnosticEcuResetClass, DiagnosticIoControlClass, DiagnosticReadDTCInformationClass, DiagnosticReadDataByIdentifierClass, DiagnosticReadDataByPeriodicIDClass, DiagnosticReadMemoryByAddressClass, DiagnosticReadScalingDataByIdentifierClass, DiagnosticRequestControlOfOnBoardDeviceClass, DiagnosticRequestCurrentPowertrainDataClass, DiagnosticRequestDownloadClass, DiagnosticRequestEmissionRelatedDTCClass, DiagnosticRequestEmissionRelatedDTCPermanentStatusClass, DiagnosticRequestFileTransferClass, DiagnosticRequestOnBoardMonitoringTestResultsClass, DiagnosticRequestPowertrainFreezeFrameDataClass, DiagnosticRequestUploadClass, DiagnosticRequestVehicleInfoClass, DiagnosticResponseOnEventClass, DiagnosticRoutineControlClass, DiagnosticSecurityAccessClass, DiagnosticSessionControlClass, DiagnosticTransferExitClass, DiagnosticWriteDataByIdentifierClass, DiagnosticWriteMemoryByAddressClass

▽

△

Class	***DiagnosticServiceClass*** (abstract)			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
access Permission	DiagnosticAccess Permission	0..1	ref	This represents the collection of DiagnosticAccess Permissions that allow for the execution of the referencing DiagnosticServiceClass.
access Permission Validity	DiagnosticAccess PermissionValidityEnum	1	attr	This attribute is responsible for clarifying the validity of the accessPermission reference.

Table A.241: DiagnosticServiceClass

Class	**DiagnosticServiceDataMapping**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::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*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
diagnosticData Element	DiagnosticDataElement	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.
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.

Table A.242: 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*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	DiagnosticClearDiagnosticInformation, DiagnosticClearResetEmissionRelatedInfo, DiagnosticComControl, DiagnosticControlDTCSetting, *DiagnosticDataByIdentifier*, DiagnosticDynamicallyDefineDataIdentifier, DiagnosticEcuReset, DiagnosticIOControl, *DiagnosticMemoryByAddress*, DiagnosticReadDTCInformation, DiagnosticReadDataByPeriodicID, DiagnosticRequestControlOfOnBoardDevice, DiagnosticRequestCurrentPowertrainData, DiagnosticRequestEmissionRelatedDTC, DiagnosticRequestEmissionRelatedDTCPermanentStatus, DiagnosticRequestFileTransfer, DiagnosticRequestOnBoardMonitoringTestResults, DiagnosticRequestPowertrainFreezeFrameData, DiagnosticRequestVehicleInfo, DiagnosticResponseOnEvent, DiagnosticRoutineControl, DiagnosticSecurityAccess, DiagnosticSessionControl			
Attribute	***Type***	***Mul.***	***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..

▽

△

Class	***DiagnosticServiceInstance*** (abstract)			
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.243: DiagnosticServiceInstance

Class	**DiagnosticServiceSwMapping**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
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,
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 hiearchy (under possible consideration of the root SoftwareComposition)
serviceInstance	DiagnosticService Instance	0..1	ref	This represents the service instance that needs to be considered in this diagnostics service mapping.

Table A.244: 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*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Attribute	***Type***	***Mul.***	***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, variation Point.shortLabel vh.latestBindingTime=postBuild

△

Class	**DiagnosticServiceTable**			
ecuInstance	EcuInstance	0..1	ref	This represents the applicable EcuInstance for this DiagnosticServiceTable.
protocolKind	NameToken	1	attr	This identifies the applicable protocol.
serviceInstance	DiagnosticService Instance	1..*	ref	This represents the collection of DiagnosticService Instances to be considered in the scope of this Diagnostic ServiceTable,

Table A.245: DiagnosticServiceTable

Class	**DiagnosticStartRoutine**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
Note	This represents the ability to start a diagnostic routine.			
Base	*ARObject*, *DiagnosticRoutineSubfunction*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
request	DiagnosticParameter	*	aggr	This represents the request parameters.
response	DiagnosticParameter	*	aggr	This represents the response parameters.

Table A.246: DiagnosticStartRoutine

Class	**DiagnosticStopRoutine**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
Note	This represents the ability to stop a diagnostic routine.			
Base	*ARObject*, *DiagnosticRoutineSubfunction*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
request	DiagnosticParameter	*	aggr	This represents the request parameters.
response	DiagnosticParameter	*	aggr	This represents the response parameters.

Table A.247: DiagnosticStopRoutine

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*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
event	DiagnosticEvent	0..1	ref	This attribute represents the diagnostic event that is related to the diagnostic test result.
monitored Identifier	Diagnostic MeasurementIdentifier	1	ref	This attribute represents the related diagnostic monitored identifier.
testIdentifier	DiagnosticTestIdentifier	1	aggr	This attribute represents the applicable test identifier.
updateKind	DiagnosticTestResult UpdateEnum	1	attr	This attribute controls the update behavior of the enclosing DiagnosticTestResult.

Table A.248: 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*			
Attribute	***Type***	***Mul.***	***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.249: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dtc	DiagnosticTroubleCode	*	ref	This represents the collection of DiagnosticTroubleCodes defined by this DiagnosticTroubleCodeGroup. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=dtc, variationPoint.shortLabel vh.latestBindingTime=postBuild
groupNumber	PositiveInteger	1	attr	This represents the base number of the DTC group. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table A.250: 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*			
Attribute	***Type***	***Mul.***	***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.

△

Class	**DiagnosticTroubleCodeJ1939**			
node	DiagnosticJ1939Node	0..1	ref	This represents the related DiagnosticJ1939Node.
spn	DiagnosticJ1939Spn	0..1	ref	This represents the releated SPN.

Table A.251: DiagnosticTroubleCodeJ1939

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
considerPto Status	Boolean	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.
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.
obdDTCValue	PositiveInteger	0..1	attr	Unique Diagnostic Trouble Code value for OBD. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table A.252: 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*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
aging	DiagnosticAging	0..1	ref	Reference to an aging algorithm in case that an aging/unlearning of the event is allowed.
agingAllowed	Boolean	0..1	attr	This represents the decision whether aging is allowed for this DiagnosticTroubleCodeProps.
environment CaptureTo Reporting	EnvironmentCaptureTo ReportingEnum	0..1	attr	This attribute determines the point in time, when the data actually is captured.
extendedData Record	DiagnosticExtended DataRecord	*	ref	Defines the links to an extended data class sampler. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	**DiagnosticTroubleCodeProps**			
freezeFrame	DiagnosticFreezeFrame	*	ref	Define the links to a freeze frame class sampler. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
freezeFrame Content	DiagnosticDataIdentifier Set	0..1	ref	This represents the freeze frame layout as a set of DIDs.
freezeFrame ContentWwh Obd	DiagnosticDataIdentifier Set	0..1	ref	This reefrence identifies the layout of the WWH-OBD freeze frame.
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
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.
memory Destination	DiagnosticMemory Destination	*	ref	The event destination assigns events to none, one or multiple origins.
priority	PositiveInteger	1	attr	Priority of the event, in view of full event buffer. A lower value means higher priority.
significance	DiagnosticSignificance Enum	0..1	attr	Significance of the event, which indicates additional information concerning fault classification and resolution.

Table A.253: DiagnosticTroubleCodeProps

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 asssignedPort. 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dataLength	PositiveInteger	0..1	attr	This attribute is applicable only if the ServiceNeed is aggregated within 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 controls whether the data can be read and written or whether it is to be handled read-only.
didNumber	PositiveInteger	0..1	attr	This represents a Data identifier for the diagnostic value. This allows to predefine the DID number if the responsible function developer has received a particular requirement from the OEM or from a standardization body.
fixedLength	Boolean	0..1	attr	This attribute controls whether the data length of the data is fixed.

▽

△

Class	**DiagnosticValueNeeds**			
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.254: DiagnosticValueNeeds

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
writeClass	DiagnosticWriteDataBy IdentifierClass	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 DiagnosticWriteDataBy Identifier in the given context.

Table A.255: DiagnosticWriteDataByIdentifier

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.256: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.257: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.258: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.259: 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			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.260: 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*			
Attribute	***Type***	***Mul.***	***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. Only PduTriggerings of Dcm IPdus shall be referenced here.

Table A.261: 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*			
Attribute	***Type***	***Mul.***	***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.262: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
defList	DefList	0..1	aggr	This represents a definition list in the documentation block. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=40
figure	MIFigure	0..1	aggr	This represents a figure in the documentation block. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=70
formula	MIFormula	0..1	aggr	This is a formula in the definition block. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=60
labeledList	LabeledList	0..1	aggr	This represents a labeled list. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=50
list	List	0..1	aggr	This represents numbered or unnumbered list. **Stereotypes:** atpVariation **Tags:** 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:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=80

▽

△

Class	«atpMixed» **DocumentationBlock**			
p	MultiLanguage Paragraph	0..1	aggr	This is one particular paragraph. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=10
structuredReq	StructuredReq	0..1	aggr	This aggregation supports structured requirements embedded in a documentation block. **Stereotypes:** atpVariation **Tags:** 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:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=90
verbatim	MultiLanguageVerbatim	0..1	aggr	This represents one particular verbatim text. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=20

Table A.263: 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*			
Attribute	***Type***	***Mul.***	***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.
identifiable	Identifiable	0..1	ref	This is an identifiable object which is part of the context of the documentation.

Table A.264: DocumentationContext

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dynamicPart Alternative	DynamicPartAlternative	1..*	aggr	Com IPdu alternatives that are transmitted in the Dynamic Part of the MultiplexedIPdu.

Table A.265: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
iPdu	ISignalIPdu	1	ref	Reference to a Com IPdu which is routed to the IPduM module and is combined to a multiplexedPdu.
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.
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.266: DynamicPartAlternative

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*			
Attribute	***Type***	***Mul.***	***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.267: 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*			
Attribute	***Type***	***Mul.***	***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.
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.268: 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, must be provided as context information.			
Base	*ARObject*, *EOCExecutableEntityRefAbstract*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***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.
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.269: EOCExecutableEntityRef

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*			
Attribute	***Type***	***Mul.***	***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.
nestedEle- ment (ordered)	EOCExecutableEntity RefAbstract	1..*	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.270: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
hardware Element	HwDescriptionEntity	*	ref	Reference from the EcuAbstractionComponentType to the description of the used HwElements.

Table A.271: 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*			
Attribute	***Type***	***Mul.***	***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.
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).

▽

△

Class	EcuInstance			
commController	Communication Controller	1..*	aggr	CommunicationControllers of the ECU.
connector	Communication Connector	*	aggr	All channels controlled by a single controller.
diagnostic Address	Integer	0..1	attr	An ECU specific ID for responses of diagnostic routines.
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.
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.
pncPrepare SleepTimer	TimeValue	0..1	attr	Time in seconds the PNC state machine shall wait in PNC_PREPARE_SLEEP.
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.
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.272: 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*			
Attribute	***Type***	***Mul.***	***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.273: EcuPartition

Class	**EcuResourceEstimation**			
Package	M2::AUTOSARTemplates::SystemTemplate::SWmapping			
Note	Resource estimations for RTE and BSW of a single ECU instance.			
Base	*ARObject*			
Attribute	***Type***	***Mul.***	***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.274: EcuResourceEstimation

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***	***Mul.***	***Kind***	***Note***
configClass	EcucConfigurationClass Enum	1	attr	Specifies the ConfigurationClass for the given ConfigurationVariant.
configVariant	EcucConfiguration VariantEnum	1	attr	Specifies the ConfigurationVariant the ConfigurationClass is specified for.

Table A.275: 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			
Attribute	***Type***	***Mul.***	***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.276: 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*, *MultilanguageReferrable*, *Referrable*			
Subclasses	*EcucAbstractExternalReferenceDef*, *EcucAbstractInternalReferenceDef*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
–	–	–	–	–

Table A.277: EcucAbstractReferenceDef

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*, *MultilanguageReferrable*, *Referrable*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
choice	EcucParamConfContainerDef	*	aggr	The choices available in a EcucChoiceContainerDef. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=shortName

Table A.278: EcucChoiceContainerDef

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**	**Mul.**	**Kind**	**Note**
multiplicityConfigClass	EcucMultiplicityConfigurationClass	*	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	1	attr	String specifying if this configuration parameter is an AUTOSAR standardized configuration parameter or if the parameter is hardware- or vendor-specific.
postBuildVariantMultiplicity	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.
postBuildVariantValue	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.

▽

△

Class	***EcucCommonAttributes*** (abstract)			
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.279: EcucCommonAttributes

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			
Attribute	***Type***	***Mul.***	***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 EcucModuleDef 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
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.280: EcucContainerDef

Class	**EcucContainerValue**			
Package	M2::AUTOSARTemplates::ECUCDescriptionTemplate			
Note	Represents a Container definition in the ECU Configuration Description.			
Base	*ARObject*, *EcucIndexableValue*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

▽

△

Class	**EcucContainerValue**			
definition	EcucContainerDef	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=definition, variationPoint.shortLabel vh.latestBindingTime=postBuild
referenceValue	EcucAbstractReference Value	*	aggr	Aggregates all References with this container. atpVariation: [RS_ECUC_00079] **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=definition, variationPoint.shortLabel vh.latestBindingTime=postBuild
subContainer	EcucContainerValue	*	aggr	Aggregates all sub-containers within this container. atpVariation: [RS_ECUC_00078] **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=definition, shortName, variation Point.shortLabel vh.latestBindingTime=postBuild

Table A.281: 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***	***Mul.***	***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	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

▽

△

Class	***EcucDefinitionElement*** (abstract)			
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
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.282: EcucDefinitionElement

Class	**EcucDestinationUriDef**			
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate			
Note	Description of an EcucDestinationUriDef that is used as target of EcucUriReferenceDefs.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
destinationUri Policy	EcucDestinationUri Policy	1	aggr	Description of the targeted EcucContainerDef.

Table A.283: EcucDestinationUriDef

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*			
Attribute	***Type***	***Mul.***	***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	1	attr	This attribute defines how the referenced target Ecuc ContainerDef is described.

▽

△

Class	EcucDestinationUriPolicy			
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.284: EcucDestinationUriPolicy

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*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
–	–	–	–	–

Table A.285: EcucFunctionNameDef

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*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
container	EcucContainerValue	1..*	aggr	Aggregates all containers that belong to this module configuration. atpVariation: [RS_ECUC_00078] **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=definition, shortName, variation Point.shortLabel vh.latestBindingTime=postBuild xml.sequenceOffset=10
definition	EcucModuleDef	1	ref	Reference to the definition of this EcucModule ConfigurationValues element. Typically, this is a vendor specific module configuration. **Tags:** xml.sequenceOffset=-10

▽

△

Class	**EcucModuleConfigurationValues**			
ecucDefEdition	RevisionLabelString	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	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 must 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.286: 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*, *EcucDefinitionElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
apiServicePrefix	CIdentifier	0..1	attr	For CDD modules this attribute holds the apiService Prefix. The shortName of the module definition of a Complex Driver is always "Cdd". Therefore for CDD modules the module apiServicePrefix is described with this attribute.
container	EcucContainerDef	1..*	aggr	Aggregates the top-level container definitions of this specific module definition. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=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.

▽

△

Class	**EcucModuleDef**			
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 Ecuc ModuleDef 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 EcucModuleDef 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.287: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.288: EcucMultiplicityConfigurationClass

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*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
parameter	EcucParameterDef	*	aggr	The parameters defined within the EcucParamConf ContainerDef. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=shortName
reference	EcucAbstractReference Def	*	aggr	The references defined within the EcucParamConf ContainerDef. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=shortName
subContainer	EcucContainerDef	*	aggr	The containers defined within the EcucParamConf ContainerDef. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=shortName

Table A.289: 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*, *MultilanguageReferrable*, *Referrable*			
Subclasses	*EcucAbstractStringParamDef*, EcucAddInfoParamDef, EcucBooleanParamDef, EcucEnumerationParamDef, EcucFloatParamDef, EcucIntegerParamDef			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
derivation	EcucDerivationSpecification	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.
symbolicNameValue	Boolean	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 "isAutoValue" 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 "isAutoValue" attribute of the respective parameter to "true". If withAuto is not present the default is "false".

Table A.290: EcucParameterDef

Class	***EcucParameterValue*** (abstract)			
Package	M2::AUTOSARTemplates::ECUCDescriptionTemplate			
Note	Common class to all types of configuration values.			
Base	*ARObject*, *EcucIndexableValue*			
Subclasses	EcucAddInfoParamValue, EcucNumericalParamValue, EcucTextualParamValue			
Attribute	***Type***	***Mul.***	***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	1	ref	Reference to the definition of this EcucParameterValue subclasses in the ECU Configuration Parameter Definition. **Tags:** xml.sequenceOffset=-10

▽

△

Class	***EcucParameterValue*** (abstract)			
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.291: EcucParameterValue

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*			
Attribute	***Type***	***Mul.***	***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.292: EcucQueryExpression

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
value	Referrable	1	ref	Specifies the destination of the reference.

Table A.293: EcucReferenceValue

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*, *EcucCommonAttributes*, *EcucDefinitionElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
destinationUri	EcucDestinationUriDef	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.294: EcucUriReferenceDef

Class	**EcucValueConfigurationClass**			
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate			
Note	Specifies the ValueConfigurationClass of a parameter/reference for each ConfigurationVariant of the EcucModuleDef.			
Base	*ARObject*, *EcucAbstractConfigurationClass*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.295: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
category	NameToken	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
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

▽

△

Class	**EndToEndDescription**			
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
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. ▽

▽

△

Class	EndToEndDescription			
				△ 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.296: EndToEndDescription

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*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
endToEnd Profile	EndToEndDescription	1	aggr	This represents the particular EndToEndDescription. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=endToEndProfile
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=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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortLabel, variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table A.297: 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.			
Base	*ARObject*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
receiver	VariableDataPrototype	*	iref	This represents the receiver. Note that 1:n communication is supported for this use case.
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.
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.

Table A.298: EndToEndProtectionVariablePrototype

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
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.
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.
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.

▽

△

Class	**EndToEndTransformationDescription**			
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 must 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 must be, due to compatibility between non-protected and E2E-protected communication, at the same position, which is before E2E header.
windowSize	PositiveInteger	0..1	attr	Size of the monitoring window for the E2E state machine.

Table A.299: EndToEndTransformationDescription

Class	«atpVariation» **EndToEndTransformationISignalProps**			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	Holds all the ISignal specific attributes for the EndToEndTransformer.			
Base	*ARObject*, *Describable*, *TransformationISignalProps*			
Attribute	***Type***	***Mul.***	***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.

Table A.300: EndToEndTransformationISignalProps

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*			
Attribute	***Type***	***Mul.***	***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.301: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
tracedFailure	TracedFailure	1..*	aggr	list of traced failures **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table A.302: ErrorTracerNeeds

Class	**EthGlobalTimeDomainProps**			
Package	M2::AUTOSARTemplates::SystemTemplate::GlobalTime::ETH			
Note	Enables the definition of Ethernet Global Time specific properties.			
Base	*ARObject*, *AbstractGlobalTimeDomainProps*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
crcFlags	EthTSynCrcFlags	0..1	aggr	Defines the fields of the message which shall be taken into account for CRC calculation and verification.
destination Physical Address	MacAddressString	0..1	attr	Defines the MAC multicast address the Ethernet time sync messages are communicated on.
fupData IDList (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.303: EthGlobalTimeDomainProps

Class	«atpVariation» **EthernetCluster**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Ethernet-specific cluster attributes. **Tags:** atp.ManifestKind=MachineManifest atp.recommendedPackage=CommunicationClusters			
Base	*ARObject*, *CollectableElement*, *CommunicationCluster*, *FibexElement*, *Identifiable*, *Multilanguage Referrable*, *PackageableElement*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
couplingPort Connection	CouplingPort Connection	*	aggr	Specification of connections between CouplingElements and EcuInstances. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=couplingPortConnection, variation Point.shortLabel vh.latestBindingTime=postBuild
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.304: EthernetCluster

Class	**EthernetCommunicationConnector**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Ethernet specific attributes to the CommunicationConnector. **Tags:** atp.ManifestKind=MachineManifest			
Base	*ARObject*, *CommunicationConnector*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
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.
network Endpoint	NetworkEndpoint	*	ref	NetworkEndpoints
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.
pncFilterData Mask	PositiveUnlimitedInteger	0..1	attr	Bit mask for Ethernet Payload used to configure the Ethernet Transceiver for partial network wakeup.

Table A.305: EthernetCommunicationConnector

Class	«atpVariation» **EthernetCommunicationController**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Ethernet specific communication port attributes.			
Base	*ARObject*, *CommunicationController*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
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.

Table A.306: EthernetCommunicationController

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. **Tags:** atp.ManifestKind=MachineManifest			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *PhysicalChannel*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
network Endpoint	NetworkEndpoint	*	aggr	Collection of NetworkEndpoints that are used in the VLan. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=shortName
soAdConfig	SoAdConfig	0..1	aggr	SoAd Configuration for one specific Physical Channel.
vlan	VlanConfig	0..1	aggr	VLAN Configuration.

Table A.307: EthernetPhysicalChannel

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*			
Attribute	***Type***	***Mul.***	***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.308: EthernetPriorityRegeneration

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*			
Attribute	***Type***	***Mul.***	***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.309: EvaluatedVariantSet

Class	**EventHandler**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Configures the outbound application endpoint a server uses to call a clients callback. Only required if the source TpPort is not dynamically assigned. If a consumed event group is referenced the configuration is only valid for this relation.			
Base	*ARObject*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
consumedEvent Group	ConsumedEventGroup	*	ref	All consumers of the event are referenced here.
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 ▽

▽

△

Class	EventHandler			
				△ 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.
routingGroup	SoAdRoutingGroup	*	ref	The ServiceDiscovery module is able to activate and deactivate the PDU routing for events.
sdServerConfig	SdServerConfig	0..1	aggr	Server configuration parameter for Service-Discovery.

Table A.310: EventHandler

Class	ExclusiveArea			
Package	M2::AUTOSARTemplates::CommonStructure::InternalBehavior			
Note	Prevents an executable entity running in the area from being preempted.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
–	–	–	–	–

Table A.311: ExclusiveArea

Class	**ExecutableEntity** (abstract)			
Package	M2::AUTOSARTemplates::CommonStructure::InternalBehavior			
Note	Abstraction of executable code.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	*BswModuleEntity*, RunnableEntity			
Attribute	**Type**	**Mul.**	**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	ExclusiveArea	*	ref	This means that the executable entity can enter/leave the referenced exclusive area through explicit API calls.
exclusiveArea NestingOrder	ExclusiveAreaNesting Order	*	ref	This represents the set of ExclusiveAreaNestingOrders recognized by this ExecutableEntity.
minimumStart Interval	TimeValue	1	attr	Specifies the time in seconds by which two consecutive starts of an ExecutableEntity are guaranteed to be separated.
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	ExclusiveArea	*	ref	The executable entity runs completely inside the referenced exclusive area.

▽

△

Class	***ExecutableEntity*** (abstract)			
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.312: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
bitPosition	PositiveInteger	1	attr	This attribute allows for defining the position of the enclosing ExecutableEntityActivationReason in the activation vector.

Table A.313: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
base Composition	CompositionSw ComponentType	0..1	ref	Specifies the composition SW-C type playing the role of a SW-C containing further SW-Cs and represents the scope of the Execution Order Constraint.
executionOrder ConstraintType	ExecutionOrder ConstraintTypeEnum	0..1	attr	Specifies the specific type of ExecutionOrderConstraint.
ignoreOrder Allowed	Boolean	0..1	attr	Controls whether the order of execution specified by this constraint can be intentionally ignored (TRUE), or shall be respected (FALSE).

▽

△

Class	**ExecutionOrderConstraint**			
isEvent	Boolean	0..1	attr	Indicates whether the ExecutionOrderConstraint is only referring to Executable Entities (FALSE) or only to RTE and/or BSW Events (TRUE).
orderedElement	EOCExecutableEntity RefAbstract	1..*	aggr	The list of references to ExecutableEntities which shall be ordered.
permitMultiple ReferencesTo EE	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.314: ExecutionOrderConstraint

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*			
Attribute	***Type***	***Mul.***	***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). **Tags:** atp.Status=shallBecomeMandatory 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

Table A.315: ExternalTriggeringPoint

Class	**FMAttributeDef**			
Package	M2::AUTOSARTemplates::FeatureModelTemplate			
Note	This metaclass represents the ability to define attributes for a feature.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
defaultValue	Numerical	0..1	attr	This represents the default value of the attribute.
max	Limit	1	attr	Maximum possible value for the value of this attribute
min	Limit	1	attr	Minimum possible value for the value of this attribute

Table A.316: FMAttributeDef

Class	**FMAttributeValue**			
Package	M2::AUTOSARTemplates::FeatureModelTemplate			
Note	This defines a value for the attribute that is referred to in the role definition.			
Base	*ARObject*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
definition	FMAttributeDef	1	ref	This refers to the definition of this attribute.
value	Numerical	1	attr	This represents the value of this attribute.

Table A.317: FMAttributeValue

Class	**FMFeature**			
Package	M2::AUTOSARTemplates::FeatureModelTemplate			
Note	A FMFeature describes an essential characteristic of a product. Each FMFeature is contained in exactly one FMFeatureModel. **Tags:** atp.recommendedPackage=FMFeatureModels			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
attributeDef	FMAttributeDef	*	aggr	This defines the attributes of the given feature.
decomposition	FMFeature Decomposition	*	aggr	Lists the sub-features of a feature.
maximum IntendedBinding Time	BindingTimeEnum	0..1	attr	Defines an upper bound for the binding time of the variation points that are associated with the FMFeature. This attribute is meant as a hint for the development process.
minimum IntendedBinding Time	BindingTimeEnum	0..1	attr	Defines a lower bound for the binding time of the variation points that are associated with the FMFeature. This attribute is meant as a hint for the development process.
relation	FMFeatureRelation	*	aggr	Defines relations for FMFeatures, for example dependencies on other FMFeatures, or conflicts with other FMFeatures. A FMFeature can only be part of a FMFeatureSelectionSet if all its relations are fulfilled.
restriction	FMFeatureRestriction	*	aggr	Defines restrictions for FMFeatures. A FMFeature can only be part of a FMFeatureSelectionSet if at least one of its restrictions evaluates to true.

Table A.318: FMFeature

▽

Class	**FMFeatureDecomposition**			
Package	M2::AUTOSARTemplates::FeatureModelTemplate			
Note	A FMFeatureDecomposition describes dependencies between a list of features and their parent feature (i.e., the FMFeature that aggregates the FMFeatureDecomposition). The kind of dependency is defined by the attribute category.			
Base	*ARObject*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
category	CategoryString	1	attr	The category of a FMFeatureDecomposition defines the type of dependency that is defined by the FMFeature Decomposition. There are four different categories: MANDATORYFEATURE, OPTIONALFEATURE, ALTERNATIVEFEATURE, and MULTIPLEFEATURE.

▽

△

Class	FMFeatureDecomposition			
feature	FMFeature	1..*	ref	The features that are affected by the dependency defined by the FMFeatureDecomposition.
max	PositiveInteger	0..1	attr	For a dependency of category MULTIPLEFEATURE, this defines the maximum number of features allowed.
min	PositiveInteger	0..1	attr	For a dependency of category MULTIPLEFEATURE, this defines the minimum number of features allowed.

Table A.319: FMFeatureDecomposition

Class	FMFeatureModel			
Package	M2::AUTOSARTemplates::FeatureModelTemplate			
Note	A Feature model describes the features of a product line and their dependencies. Feature models are an optional part of an AUTOSAR model. **Tags:** atp.recommendedPackage=FMFeatureModels			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
feature	FMFeature	*	ref	"feature" holds the list of features of the feature model. No FMFeature may be contained twice in this list. Also, each FMFeature may be contained on only one feature model. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=feature
root	FMFeature	0..1	ref	The features of a feature model define a tree. The attribute root points to the root of this tree.

Table A.320: FMFeatureModel

Class	FMFeatureRelation			
Package	M2::AUTOSARTemplates::FeatureModelTemplate			
Note	Defines relations for FMFeatures, for example dependencies on other FMFeatures, or conflicts with other FMFeatures. A FMFeature can only be part of a FMFeatureSelectionSet if all its relations are fulfilled.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
feature	FMFeature	1..*	ref	The FMFeature that is targeted by this FMFeature Relation.

Table A.321: FMFeatureRelation

Class	FMFeatureSelection			
Package	M2::AUTOSARTemplates::FeatureModelTemplate			
Note	A FMFeatureSelection represents the state of a particular FMFeature within a FMFeatureSelectionSet.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
attributeValue	FMAttributeValue	*	aggr	This defines a value for the attribute that is referred to in the role definition. Note that a FMFeatureSelection cannot include two FMAttributeValues that refer to the same FMAttributeDef in the role definition. **Tags:** xml.sequenceOffset=50

▽

△

Class	**FMFeatureSelection**			
feature	FMFeature	1	ref	The FMFeature whose state is defined by this FMFeature Selection. **Tags:** xml.sequenceOffset=10
maximum SelectedBinding Time	BindingTimeEnum	0..1	attr	Defines an upper bound for the binding time of the variation points that are associated with the FMFeature, and refines its maximumIntendedBindingTime. This attribute is meant as a hint for the development process. **Tags:** xml.sequenceOffset=40
minimum SelectedBinding Time	BindingTimeEnum	0..1	attr	Defines a lower bound for the binding time of the variation points that are associated with the FMFeature, and refines its minimumIntendedBindingTime. This attribute is meant as a hint for the development process. **Tags:** xml.sequenceOffset=30
state	FMFeatureSelection State	1	attr	Defines how the FMFeature that is described by this FMFeatureSelection contributes to the FMFeature SelectionSet. A FMFeature may have the state selected, deselected or undecided. **Tags:** xml.sequenceOffset=20

Table A.322: FMFeatureSelection

Class	**FMFeatureSelectionSet**			
Package	M2::AUTOSARTemplates::FeatureModelTemplate			
Note	A FMFeatureSelectionSet is a set of FMFeatures that describes a specific product. **Tags:** atp.recommendedPackage=FMFeatureModelSelectionSets			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
featureModel	FMFeatureModel	*	ref	All FMFeatures in this FMFeatureSelectionSet shall be part of the referenced FMFeatureModel.
include	FMFeatureSelectionSet	*	ref	Each FMFeatureSelectionSet may include one or more FMFeatureSelectionSets. This establishes a hierarchy among FMFeatureSelectionSets. See constr_5003 and constr_5025 for details.
selection	FMFeatureSelection	*	aggr	The set of FMFeatureSelections of this FMFeature SelectionSet.

Table A.323: FMFeatureSelectionSet

Class	«atpMixedString» ***FMFormulaByFeaturesAndAttributes*** (abstract)
Package	M2::AUTOSARTemplates::FeatureModelTemplate
Note	An expression that has the syntax of the AUTOSAR formula language but uses only references to features or feature attributes (not system constants) as operands.
Base	*ARObject*, *FormulaExpression*
Subclasses	FMConditionByFeaturesAndAttributes

▽

△

Class	«atpMixedString» ***FMFormulaByFeaturesAndAttributes*** (abstract)			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
attribute	FMAttributeDef	1	ref	An expression of type FMFormulaByFeaturesAnd Attributes may refer to attributes of FMFeatures.
feature	FMFeature	1	ref	An expression of type FMFormulaByFeaturesAnd Attributes may refer to FMFeatures.

Table A.324: FMFormulaByFeaturesAndAttributes

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*, CouplingElement, EcuInstance, *Frame*, Gateway, GlobalTimeDomain, ISignal, ISignalGroup, ISignalIPduGroup, NmConfig, *Pdu*, PdurIPduGroup, Secure CommunicationPropsSet, SoAdRoutingGroup, *TpConfig*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.325: 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*			
Attribute	***Type***	***Mul.***	***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

▽

△

Class	**FlatInstanceDescriptor**			
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.
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

Table A.326: FlatInstanceDescriptor

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*			
Attribute	***Type***	***Mul.***	***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 CommunicationConnector.
multicast	TpAddress	0..1	ref	TP address for 1:n connections.

▽

△

Class	**FlexrayArTpConnection**			
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 CommunicationConnector.
source	FlexrayArTpNode	1	ref	The source of the TP connection.
target	FlexrayArTpNode	1..*	ref	The target of the TP connection.

Table A.327: FlexrayArTpConnection

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*			
Attribute	***Type***	***Mul.***	***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.
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.

▽

△

Class	«atpVariation» **FlexrayCluster**			
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.
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:bitDuration 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.

▽

△

Class	«atpVariation» **FlexrayCluster**			
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.328: FlexrayCluster

Class	**FlexrayFrameTriggering**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Flexray::FlexrayCommunication			
Note	FlexRay specific attributes to the FrameTriggering			
Base	*ARObject*, *FrameTriggering*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***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.329: FlexrayFrameTriggering

Class	**FlexrayPhysicalChannel**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology			
Note	FlexRay specific attributes to the physicalChannel			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *PhysicalChannel*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
channelName	FlexrayChannelName	1	attr	Name of the channel (Channel A or Channel B).

Table A.330: 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*			
Attribute	***Type***	***Mul.***	***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.
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 rx PduPool 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 tx PduPool holds the actually sent NPdus. In case this connection is applied to the receiver the txPduPool holds the actually received NPdus.

Table A.331: 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*

▽

△

Class	***Frame*** (abstract)			
Subclasses	*AbstractEthernetFrame*, CanFrame, FlexrayFrame, *LinFrame*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
frameLength	Integer	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.
pduToFrame Mapping	PduToFrameMapping	*	aggr	A frames layout as a sequence of Pdus. atpVariation: The content of a frame can be variable. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild

Table A.332: Frame

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.333: 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			
Attribute	***Type***	***Mul.***	***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:** atpVariation **Tags:** vh.latestBindingTime=postBuild

Table A.334: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.335: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
pduTriggering	PduTriggering	*	ref	Reference to PduTriggerings that are connected to each other by a GeneralPurposeConnection.

Table A.336: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.337: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.338: 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*			
Attribute	***Type***	***Mul.***	***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:** atpVariation **Tags:** 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:** atpVariation **Tags:** 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:** atpVariation **Tags:** 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:** atpVariation **Tags:** vh.latestBindingTime=postBuild
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.
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:** atpVariation **Tags:** vh.latestBindingTime=postBuild
slave	GlobalTimeSlave	*	aggr	This represents the collections of slaves of the Global TimeDomain. A GlobalTimeDomain may have no GlobalTime Domain.slaves, e.g. when it propagates its time directly to sub domains. **Stereotypes:** atpVariation **Tags:** 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.339: 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*			
Attribute	***Type***	***Mul.***	***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.340: 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*			
Attribute	***Type***	***Mul.***	***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.341: 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			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
communication Connector	Communication Connector	1	ref	The GlobalTimeMaster is bound to the Communication Connector.
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.342: 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			
Attribute	***Type***	***Mul.***	***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.
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.343: GlobalTimeSlave

Enumeration	**HandleInvalidEnum**
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication
Note	Strategies of handling the reception of invalidValue.
Literal	***Description***
dontInvalidate	Invalidation is switched off. **Tags:** atp.EnumerationValue=0
external Replacement	Replace a received invalidValue. The replacement value is sourced from the externalReplacement. **Tags:** atp.EnumerationValue=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.EnumerationValue=2
replace	Replace a received invalidValue. The replacement value is specified by the initValue. **Tags:** atp.EnumerationValue=3

Table A.344: HandleInvalidEnum

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 hw AttributeEnumerationLiterals specify the available literals.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
hwAttribute Literal	HwAttributeLiteralDef	*	aggr	The available EnumerationLiterals of the Enumeration definition. Only applicable if the category of the Hw AttributeDef equals Enumeration.

▽

△

Class	**HwAttributeDef**			
isRequired	Boolean	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.345: HwAttributeDef

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***	***Mul.***	***Kind***	***Note***
hwAttribute Value	HwAttributeValue	*	aggr	This aggregation represents a particular hardware attribute value. **Stereotypes:** atpVariation **Tags:** 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 HwTypes can not be redefined and therefore shall not have a hwType reference.

Table A.346: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
hwElement Connection	HwElementConnector	*	aggr	This represents one particular connection between two hardware elements. **Stereotypes:** atpVariation **Tags:** 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:** atpVariation **Tags:** 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:** atpVariation **Tags:** vh.latestBindingTime=systemDesignTime xml.sequenceOffset=70

Table A.347: HwElement

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.348: HwType

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*, *PackageableElement*, *Pdu*, *Referrable*			
Subclasses	ContainerIPdu, DcmIPdu, GeneralPurposeIPdu, ISignalIPdu, J1939DcmIPdu, MultiplexedIPdu, NPdu, SecuredIPdu, UserDefinedIPdu			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
containedIPdu Props	ContainedIPduProps	0..1	aggr	Defines whether this IPdu may be collected inside a ContainerIPdu.

Table A.349: IPdu

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*			
Attribute	***Type***	***Mul.***	***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.

▽

△

Class	**IPduPort**			
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.350: IPduPort

Class	**IPv6ExtHeaderFilterList**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication			
Note	White list for the filtering of IPv6 extension headers.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
allowedIPv6Ext Header	PositiveInteger	1..*	attr	IPv6 Extension Header type allowed by this filter.

Table A.351: 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 must be created for each SystemSignal contained in the SystemSignalGroup. **Tags:** atp.recommendedPackage=ISignals			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Attribute	*Type*	*Mul.*	*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, variation Point.shortLabel vh.latestBindingTime=codeGenerationTime

▽

△

Class	**ISignal**			
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.
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.
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.
length	Integer	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.
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. ▽

▽

△

Class	**ISignal**			
				△ 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.
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.

Table A.352: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
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=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.

Table A.353: 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*			
Attribute	***Type***	***Mul.***	***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:** atpVariation **Tags:** vh.latestBindingTime=postBuild
iSignalToPdu Mapping	ISignalToIPduMapping	*	aggr	Definition of SignalToIPduMappings included in the Signal IPdu. atpVariation: The content of a PDU can be variable. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
pduCounter	SignalIPduCounter	0..1	aggr	An included Pdu counter is used to ensure that a sequence of Pdus is maintained. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
pduReplication	SignalIPduReplication	0..1	aggr	Pdu Replication is a form of redundancy where the data content of one ISignalIPdu (source) is transmitted inside a set of replica ISignalIPdus. These ISignalIPdus (copies) have different Pdu IDs, identical PduCounters, identical data content and are transmitted with the same frequency. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
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.354: 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*			
Attribute	***Type***	***Mul.***	***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).

▽

△

Class	**ISignalIPduGroup**			
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:** atpVariation **Tags:** vh.latestBindingTime=postBuild
nmPdu	NmPdu	*	ref	Reference to a set of NmPdus with NmUserData, which are contained in the ISignalIPduGroup. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild

Table A.355: 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*			
Attribute	***Type***	***Mul.***	***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.356: 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*			
Attribute	***Type***	***Mul.***	***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.

▽

△

Class	**ISignalPort**			
firstTimeout	TimeValue	0..1	attr	Optional first timeout value in seconds for the reception of the ISignal.
timeout	TimeValue	0..1	attr	Optional timeout value in seconds for the reception of the ISignal. In case 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 Receiver ComSpec, in this case the timeout value in ReceiverCom Spec override this optional timeout specification.

Table A.357: 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*			
Attribute	***Type***	***Mul.***	***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 ISignalGroup 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.
startPosition	Integer	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 packingByte Order attribute). In AUTOSAR the bit counting is always ▽

▽

△

Class	**ISignalToIPduMapping**			
				△ 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	Integer	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.358: 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*			
Attribute	***Type***	***Mul.***	***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.359: ISignalTriggering

Class	***Identifiable*** (abstract)			
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::Identifiable			
Note	Instances of this class can be referred to by their identifier (within the namespace borders). In addition to this, Identifiables are objects which contribute significantly to the overall structure of an AUTOSAR description. In particular, Identifiables might contain Identifiables.			
Base	*ARObject*, *MultilanguageReferrable*, *Referrable*			
Subclasses	ARPackage, *AbstractEvent*, *AbstractImplementationDataTypeElement*, *AbstractServiceInstance*, ApplicationEndpoint, ApplicationError, ApplicationPartitionToEcuPartitionMapping, AsynchronousServer CallResultPoint, *AtpBlueprint*, *AtpBlueprintable*, *AtpClassifier*, *AtpFeature*, AutosarOperationArgument Instance, AutosarVariableInstance, BswInternalTriggeringPoint, BswModuleDependency, *BuildAction Entity*, BuildActionEnvironment, CanTpAddress, CanTpChannel, CanTpNode, Chapter, ClassContent Conditional, ClientIdDefinition, ClientServerOperation, Code, *CollectableElement*, ComManagement Mapping, *CommConnectorPort*, *CommunicationConnector*, *CommunicationController*, Compiler, ConsistencyNeeds, ConsumedEventGroup, CouplingPort, *CouplingPortStructuralElement*, *Crypto ServiceMapping*, DataPrototypeGroup, DataTransformation, DependencyOnArtifact, *DiagEvent DebounceAlgorithm*, DiagnosticConnectedIndicator, DiagnosticDataElement, DiagnosticFunctionInhibit Source, DiagnosticMasterToSlaveEventMapping, *DiagnosticRoutineSubfunction*, DoIpLogicAddress, EC UMapping, *EOCExecutableEntityRefAbstract*, EcuPartition, EcucContainerValue, *EcucDefinition Element*, EcucDestinationUriDef, EcucEnumerationLiteralDef, EcucQuery, EcucValidationCondition, End ToEndProtection, 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, IPv6Ext HeaderFilterList, ISignalToIPduMapping, ISignalTriggering, *IdentCaption*, InternalTriggeringPoint, J1939 SharedAddressCluster, J1939TpNode, Keyword, LifeCycleState, LinScheduleTable, LinTpNode, Linker, MacMulticastGroup, McDataInstance, MemorySection, ModeDeclaration, ModeDeclarationMapping, ModeSwitchPoint, NetworkEndpoint, *NmCluster*, NmEcu, *NmNode*, NvBlockDescriptor, *Packageable Element*, ParameterAccess, PduToFrameMapping, PduTriggering, PerInstanceMemory, *Physical Channel*, PortGroup, *PortInterfaceMapping*, PossibleErrorReaction, ResourceConsumption, RootSw CompositionPrototype, RptComponent, RptContainer, RptExecutableEntity, RptExecutableEntityEvent, RptExecutionContext, RptProfile, RptServicePoint, RunnableEntityGroup, *SdgAttribute*, SdgClass, SecureCommunicationAuthenticationProps, SecureCommunicationFreshnessProps, *ServerCallPoint*, *ServiceNeeds*, SocketAddress, SomeipTpChannel, *SpecElementReference*, *StackUsage*, Structured Req, SwGenericAxisParamType, SwServiceArg, SwcServiceDependency, SwcToApplicationPartition Mapping, SwcToEcuMapping, SwcToImplMapping, SystemMapping, TcpOptionFilterList, Timing Condition, *TimingConstraint*, *TimingDescription*, TimingExtensionResource, TimingModeInstance, Tls CryptoCipherSuite, Topic1, TpAddress, TraceableText, *TracedFailure*, *TransformationProps*, TransformationTechnology, Trigger, VariableAccess, VariationPointProxy, ViewMap, VlanConfig, Wait Point			
Attribute	***Type***	***Mul.***	***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! 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

▽

△

Class	***Identifiable*** (abstract)			
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
adminData	AdminData	0..1	aggr	This represents the administrative data for the identifiable object. **Tags:** 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
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
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.360: 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

▽

△

Primitive	**Identifier**			
Attribute	***Datatype***	***Mul.***	***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.361: 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			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
buildAction Manifest	BuildActionManifest	0..1	ref	A manifest specifying the intended build actions for the software delivered with this implementation. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=codeGenerationTime
codeDescriptor	Code	1..*	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:** atpVariation **Tags:** 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	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:** atpVariation **Tags:** 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:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
resource Consumption	ResourceConsumption	1	aggr	All static and dynamic resources for each implementation are described within the ResourceConsumption class. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=shortName
swVersion	RevisionLabelString	1	attr	Software version of this implementation. The numbering contains three levels (like major, minor, patch), its values are vendor specific.
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.
usedCode Generator	String	0..1	attr	Optional: code generator used.
vendorId	PositiveInteger	1	attr	Vendor ID of this Implementation according to the AUTOSAR vendor list

Table A.362: 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*			
Attribute	***Type***	***Mul.***	***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. **Tags:** atp.Status=draft
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:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

▽

△

Class	**ImplementationDataType**			
symbolProps	SymbolProps	0..1	aggr	This represents the SymbolProps for the Implementation DataType. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=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.363: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
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.
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. **Tags:** atp.Status=draft
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:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

▽

△

Class	**ImplementationDataTypeElement**			
swDataDef Props	SwDataDefProps	0..1	aggr	The properties of this ImplementationDataTypeElement.

Table A.364: ImplementationDataTypeElement

Class	**ImplementationDataTypeSubElementRef**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	This meta-class represents the specialization of SubElementMapping with respect to Implementation DataTypes.			
Base	*ARObject*, *SubElementRef*			
Attribute	***Type***	***Mul.***	***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.365: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
context	ParameterData Prototype	1	ref	The context for the referred element. **Tags:** xml.sequenceOffset=20
target	ImplementationData TypeElement	1	ref	The referred data element. **Tags:** xml.sequenceOffset=30

Table A.366: ImplementationElementInParameterInstanceRef

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*			
Attribute	***Type***	***Mul.***	***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.367: 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*, *MultilanguageReferrable*, *RTEEvent*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.368: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
parameter Instance	AutosarParameterRef	0..1	aggr	This is the particular ParameterDataPrototypes on which the swDataDefProps shall be applied.
swDataDef Props	SwDataDefProps	1	aggr	These are the particular data definition properties which shall be applied
variableInstance	AutosarVariableRef	0..1	aggr	This is the particular VariableDataPrototypes on which the swDataDefProps shall be applied.

Table A.369: InstantiationDataDefProps

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 supports then an instance specific timing.			
Base	*ARObject*, *InstantiationRTEEventProps*			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
period	TimeValue	1	attr	This attribute represents the value of the refined activation period.

Table A.370: 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			
Attribute	*Type*	*Mul.*	*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 SwComponentPrototypes of the same SwComponent Type. 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=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
constantValue Mapping	ConstantSpecification MappingSet	*	ref	Reference to the ConstanSpecificationMapping 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 ▽

▽

△

Class	***InternalBehavior*** (abstract)			
				△ conditional existence of RunnableEntities or BswModule Entities. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, 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=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
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=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table A.371: InternalBehavior

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
swImplPolicy	SwImplPolicyEnum	0..1	attr	This attribute, when set to value queued, allows for a queued processing of Triggers.

Table A.372: InternalTriggeringPoint

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*

▽

△

Class	**InvalidationPolicy**			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dataElement	VariableDataPrototype	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.373: InvalidationPolicy

Class	**Ipv4Configuration**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Internet Protocol version 4 (IPv4) configuration.			
Base	*ARObject*, *NetworkEndpointAddress*			
Attribute	***Type***	***Mul.***	***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 ipv4Address Source 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.374: Ipv4Configuration

Class	**Ipv6Configuration**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Internet Protocol version 6 (IPv6) configuration.			
Base	*ARObject*, *NetworkEndpointAddress*			
Attribute	***Type***	***Mul.***	***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.
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

▽

△

Class	**Ipv6Configuration**			
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 ipv6Address Source 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.375: Ipv6Configuration

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*			
Attribute	***Type***	***Mul.***	***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.376: 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*			
Attribute	***Type***	***Mul.***	***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.

Table A.377: 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*			
Attribute	*Type*	*Mul.*	*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.378: 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*			
Attribute	*Type*	*Mul.*	*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.379: J1939DcmIPdu

Class	**J1939NmCluster**			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	J1939 specific NmCluster attributes			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *NmCluster*, *Referrable*			
Attribute	*Type*	*Mul.*	*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.380: J1939NmCluster

Class	**J1939NmNode**			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	J1939 specific NM Node attributes.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *NmNode*, *Referrable*			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
nodeName	J1939NodeName	0..1	aggr	NodeName configuration

Table A.381: J1939NmNode

Class	**J1939NodeName**			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	This element contains attributes to configure the J1939NmNode NAME.			
Base	*ARObject*			
Attribute	***Type***	***Mul.***	***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.
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.382: 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*			
Attribute	***Type***	***Mul.***	***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.383: 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*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
–	–	–	–	–

Table A.384: LParagraph

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*			
Attribute	**Type**	**Mul.**	**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
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 must be individually described in the remark.

Table A.385: 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*, *Packageable Element*, *Referrable*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**

▽

△

Class	**LifeCycleInfoSet**			
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 defaultLc State can be overwritten in LifeCycleInfo elements.
defaultPeriod Begin	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.
defaultPeriod End	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.
usedLifeCycle StateDefinition Group	LifeCycleStateDefinition Group	1	ref	This denotes the life cycle states applicable to the current life cycle info set.

Table A.386: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
arRelease Version	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
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.387: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.388: 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*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
lcState	LifeCycleState	*	aggr	Describes a single life cycle state of this life cycle state definition group.

Table A.389: 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	***Datatype***	***Mul.***	***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.390: Limit

Class	**LinCommunicationConnector**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinTopology			
Note	LIN bus specific communication connector attributes.			
Base	*ARObject*, *CommunicationConnector*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***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 (ordered)	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.

Table A.391: 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*			
Attribute	*Type*	*Mul.*	*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.392: 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*			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
responseError	ISignalTriggering	0..1	ref	This ISignal shall be taken to transport the responseError bit.

Table A.393: 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*			
Attribute	*Type*	*Mul.*	*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 Lin UnconditionalFrame shall be defined. Within a channel a LIN Frame shall be referenced by only one Frame Triggering. This allows a derivation of the identifier of a ▽

▽

△

Class	LinEventTriggeredFrame			
				△ substituted Frame. The identifier is specified in Frame Triggering 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.394: LinEventTriggeredFrame

Class	LinFrameTriggering			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	LIN specific attributes to the FrameTriggering			
Base	*ARObject*, *FrameTriggering*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	**Type**	**Mul.**	**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.395: 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*, *Multilanguage Referrable*, *Referrable*			
Attribute	**Type**	**Mul.**	**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.

▽

△

Class	«atpVariation» **LinMaster**			
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.396: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
frame	LinFrame	1	ref	Reference to a Frame that is processed by the slave node.
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.397: LinOrderedConfigurableFrame

Class	**LinPhysicalChannel**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinTopology			
Note	LIN specific attributes to the physicalChannel			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *PhysicalChannel*, *Referrable*			
Attribute	***Type***	***Mul.***	***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 must be variable, too. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild

Table A.398: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

▽

△

Class	**LinScheduleTable**			
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.399: 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*			
Attribute	***Type***	***Mul.***	***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.400: LinSlave

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*			
Attribute	***Type***	***Mul.***	***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 ▽

▽

△

Class	**LinSporadicFrame**			
				△ 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.401: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.402: LinUnconditionalFrame

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
rteEvent	RTEEvent	1..*	iref	The RTE event used to receive the data via this buffer.
variableAccess	VariableAccess	1..*	iref	The VariableAccess for which the data buffer is used.

Table A.403: 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. ▽

▽

△

Class	**McDataInstance**			
	△ • 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*			
Attribute	***Type***	***Mul.***	***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	ImplementationElement InParameterInstance Ref	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.
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.

▽

△

Class	**McDataInstance**			
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:** atpVariation **Tags:** 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.404: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
defCalprmSet	McFunctionDataRefSet	0..1	aggr	Refers to the set of adjustable data (= calibration parameters) defined in this function. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=variationPoint.shortLabel 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=variationPoint.shortLabel 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=variationPoint.shortLabel xml.sequenceOffset=50

▽

△

Class	**McFunction**			
out Measurement Set	McFunctionDataRefSet	0..1	aggr	Refers to the set of measurable output data from this function. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=variationPoint.shortLabel 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=variationPoint.shortLabel 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.405: 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*			
Attribute	***Type***	***Mul.***	***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. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=flatMapEntry 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. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=mcDataInstance xml.sequenceOffset=20

Table A.406: 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*			
Attribute	***Type***	***Mul.***	***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=variationPoint.shortLabel 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=variationPoint.shortLabel 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.407: 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*			
Attribute	***Type***	***Mul.***	***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. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=flatMapEntry xml.sequenceOffset=50

▽

△

Class	«atpVariation» **McGroupDataRefSet**			
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. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=mcDataInstance xml.sequenceOffset=60

Table A.408: McGroupDataRefSet

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*			
Attribute	***Type***	***Mul.***	***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:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
mcParameter Instance	McDataInstance	*	aggr	A data instance to be used for calibration. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=postBuild
mcVariable Instance	McDataInstance	*	aggr	A data instance to be used for measurement. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, 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.409: McSupportData

Class	**McSwEmulationMethodSupport**
Package	M2::AUTOSARTemplates::CommonStructure::MeasurementCalibrationSupport

▽

△

Class	**McSwEmulationMethodSupport**			
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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
category	Identifier	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
baseReference	VariableDataPrototype	0..1	ref	Refers to the base pointer in case of the double-pointered method.
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	1	attr	Assigns a name to this element. **Tags:** xml.sequenceOffset=-100

Table A.410: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
maximum ExecutionTime	MultidimensionalTime	1	aggr	The maximum measured execution time.
minimum ExecutionTime	MultidimensionalTime	1	aggr	The minimum measured execution time.
nominal ExecutionTime	MultidimensionalTime	1	aggr	The nominal measured execution time.

Table A.411: MeasuredExecutionTime

Class	**MeasuredHeapUsage**			
Package	M2::AUTOSARTemplates::CommonStructure::ResourceConsumption::HeapUsage			
Note	The heap usage has been measured.			
Base	*ARObject*, *HeapUsage*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
averageMemory Consumption	PositiveInteger	1	attr	The average heap usage measured. Unit: byte.
maximum Memory Consumption	PositiveInteger	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.412: MeasuredHeapUsage

Class	**MeasuredStackUsage**			
Package	M2::AUTOSARTemplates::CommonStructure::ResourceConsumption::StackUsage			
Note	The stack usage has been measured.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *StackUsage*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
averageMemory Consumption	PositiveInteger	1	attr	The average stack usage measured. Unit: byte.
maximum Memory Consumption	PositiveInteger	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.413: MeasuredStackUsage

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 ▽

▽

△

Class	**MemorySection**			
	△ • **[_<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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
alignment	AlignmentType	0..1	attr	The attribute describes the 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.
memClass Symbol	CIdentifier	0..1	attr	Defines a specific symbol in order to generate the compiler abstraction "memclass" code for this Memory Section. The existence of this attribute supersedes the usage of swAddrmethod.shortName for this purpose. The complete name of the "memclass" preprocessor symbol is constructed as <prefix>_<memClassSymbol> where prefix is defined in the same way as for the enclosing MemorySection. See also AUTOSAR_SWS_CompilerAbstraction SWS_COMPILER_00040.
option	Identifier	*	attr	This attribute introduces the ability to specify further intended properties of this MemorySection. 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 compiler abstraction macro INLINE. • LOCAL_INLINE - The code section is declared with the compiler abstraction macro LOCAL_INLINE In both cases (INLINE and LOCAL_INLINE) the inline expansion depends on the compiler specific implementation of these macros. Depending on this, the code section either corresponds to an actual section in memory or is put into the section of the caller. See AUTOSAR_SWS_CompilerAbstraction for more details.
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	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.414: MemorySection

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*			
Attribute	***Type***	***Mul.***	***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). **Tags:** atp.Status=shallBecomeMandatory xml.sequenceOffset=-100
modeGroup	ModeDeclarationGroup Prototype	0..1	iref	The mode declaration group that is accessed by this runnable. **Tags:** xml.typeElement=true

Table A.415: ModeAccessPoint

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. **Tags:** atp.ManifestKind=ExecutionManifest,MachineManifest
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*

▽

△

Class	ModeDeclaration			
Attribute	**Type**	**Mul.**	**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.416: ModeDeclaration

Class	ModeDeclarationGroup			
Package	M2::AUTOSARTemplates::CommonStructure::ModeDeclaration			
Note	A collection of Mode Declarations. Also, the initial mode is explicitly identified. **Tags:** atp.ManifestKind=ExecutionManifest,MachineManifest atp.recommendedPackage=ModeDeclarationGroups			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpBlueprintable*, *AtpClassifier*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
initialMode	ModeDeclaration	1	ref	The initial mode of the ModeDeclarationGroup. This mode is active before any mode switches occurred.
mode Declaration	ModeDeclaration	1..*	aggr	The ModeDeclarations collected in this ModeDeclaration Group. **Stereotypes:** atpVariation **Tags:** 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.417: 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. **Tags:** atp.ManifestKind=ExecutionManifest,MachineManifest			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	**Type**	**Mul.**	**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	1	tref	The "collection of ModeDeclarations" (= ModeDeclaration Group) supported by a component **Stereotypes:** isOfType

Table A.418: 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*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
firstModeGroup	ModeDeclarationGroup Prototype	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	1	ref	ModeDeclarationGroupPrototype to be mapped.

Table A.419: 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*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
firstMode	ModeDeclaration	1..*	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	1	ref	This represents the second ModeDeclaration of the Mode DeclarationMapping.

Table A.420: 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*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
mode Declaration Mapping	ModeDeclaration Mapping	1..*	aggr	This represents the collection of ModeDeclaration Mappings owned by the enclosing ModeDeclaration MappingSet.

Table A.421: 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*			
Attribute	***Type***	***Mul.***	***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	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.422: ModeErrorBehavior

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
modeMapping	ModeDeclarationGroup PrototypeMapping	1	aggr	Mapping of two ModeDeclarationGroupPrototypes in two different ModeInterfaces

Table A.423: ModeInterfaceMapping

Class	**ModePortAnnotation**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::ApplicationAttributes			
Note	Annotation to a port used for calibration regarding a certain ModeDeclarationGroupPrototype.			
Base	*ARObject*, *GeneralAnnotation*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
modeGroup	ModeDeclarationGroup Prototype	1	ref	The instance of annotated ModeDeclarationGroup Prototype.

Table A.424: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
implementation DataType	AbstractImplementation DataType	1	ref	This is the corresponding AbstractImplementationData Type. It shall be modeled along the idea of an "unsigned integer-like" data type.
modeGroup	ModeDeclarationGroup	1	ref	This is the corresponding ModeDeclarationGroup.

Table A.425: 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*			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
role	Identifier	1	attr	This attribute indicates which service of the NvM for the NvBlock shall be requested.
swcModeSwitch Event	SwcModeSwitchEvent	1	ref	This reference identifies the SwcModeSwitchEvent that triggers the activity.

Table A.426: 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*			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
modeGroup	ModeDeclarationGroup Prototype	1	aggr	The ModeDeclarationGroupPrototype of this mode interface.

Table A.427: ModeSwitchInterface

Class	**ModeSwitchedAckEvent**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents			
Note	The event is raised when the referenced modes have been received or an error occurs.			
Base	*ARObject*, *AbstractEvent*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *RTEEvent*, *Referrable*			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
eventSource	ModeSwitchPoint	1	ref	Mode switch point that triggers the event.

Table A.428: ModeSwitchedAckEvent

Class	**ModeSwitchedAckRequest**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	Requests acknowledgements that a mode switch has been proceeded successfully			
Base	*ARObject*			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
timeout	TimeValue	1	attr	Number of seconds before an error is reported or in case of allowed redundancy, the value is sent again.

Table A.429: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
enteredMode	ModeDeclaration	1	ref	This represents the entered model of the ModeTransition.
exitedMode	ModeDeclaration	1	ref	This represents the exited mode of the ModeTransition

Table A.430: ModeTransition

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
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 DynamicPart. 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:** atpVariation **Tags:** 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.

△

Class	**MultiplexedIPdu**			
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 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:** atpVariation **Tags:** 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 ▽

▽

△

Class	MultiplexedIPdu			
				△ 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.431: 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*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
–	–	–	–	–

Table A.432: 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). **Tags:** atp.ManifestKind=MachineManifest			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	**Type**	**Mul.**	**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.
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.433: NetworkEndpoint

Class	***NmCluster*** (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	Set of NM nodes coordinated with use of the NM algorithm. **Tags:** atp.ManifestKind=MachineManifest			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	CanNmCluster, FlexrayNmCluster, J1939NmCluster, LinNmCluster, UdpNmCluster			
Attribute	**Type**	**Mul.**	**Kind**	**Note**

▽

△

Class	***NmCluster*** (abstract)			
communication Cluster	CommunicationCluster	1	ref	Asscociation to a CommunicationCluster in the topology description.
nmChannel SleepMaster	Boolean	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.
nmNode	NmNode	*	aggr	Collection of NmNodes of the NmCluster. atpVariation: Derived, because NmNode can be variable. **Stereotypes:** atpVariation **Tags:** 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	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.

Table A.434: NmCluster

Class	**NmEcu**			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	ECU on which NM is running.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
busDependent NmEcu	BusspecificNmEcu	*	aggr	Cluster specific NmEcu attributes
ecuInstance	EcuInstance	1	ref	Asscociation 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.435: NmEcu

Class	***NmNode*** (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	The linking of NmEcus to NmClusters is realized via the NmNodes. **Tags:** atp.ManifestKind=MachineManifest			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	CanNmNode, FlexrayNmNode, J1939NmNode, UdpNmNode			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
controller	Communication Controller	1	ref	Asscociation to an CommunicationController in the topology description.
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	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. Must 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	1..*	ref	receive NM Pdu.
txNmPdu	NmPdu	*	ref	transmit NM Pdu

Table A.436: 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*			
Attribute	***Type***	***Mul.***	***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 ISignalToIPdu Mappings 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.437: NmPdu

Class	**NonqueuedReceiverComSpec**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	Communication attributes specific to non-queued receiving.			
Base	*ARObject*, *RPortComSpec*, *ReceiverComSpec*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
aliveTimeout	TimeValue	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	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 than 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	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 VariableData Prototype 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	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 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.438: NonqueuedReceiverComSpec

Class	**NonqueuedSenderComSpec**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	Communication attributes for non-queued sender/receiver communication (sender side)			
Base	*ARObject*, *PPortComSpec*, *SenderComSpec*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
initValue	ValueSpecification	1	aggr	Initial value to be sent if sender component is not yet fully initialized, but receiver needs data already.

Table A.439: NonqueuedSenderComSpec

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.440: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
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.441: NumericalOrText

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
value	Numerical	1	attr	This is the value itself. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table A.442: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
nvRamBlock Element	AutosarVariableRef	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.443: 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*			
Attribute	***Type***	***Mul.***	***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:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
constantValue Mapping	ConstantSpecification MappingSet	*	ref	Reference to the ConstanSpecificationMapping 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:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

▽

△

Class	**NvBlockDescriptor**			
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, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
nvBlockData Mapping	NvBlockDataMapping	1..*	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:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
nvBlockNeeds	NvBlockNeeds	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	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

Table A.444: 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*			
Attribute	***Type***	***Mul.***	***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.
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.445: NvBlockNeeds

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*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SwComponentType*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
nvBlock Descriptor	NvBlockDescriptor	*	aggr	Specification of the properties of exactly one NVRAM Block. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table A.446: 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			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpBlueprintable*, *AtpClassifier*, *AtpType*, *CollectableElement*, *DataInterface*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *PortInterface*, *Referrable*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
nvData	VariableDataPrototype	1..*	aggr	The VariableDataPrototype of this nv data interface.

Table A.447: NvDataInterface

Class	NvDataPortAnnotation			
Package	M2::AUTOSARTemplates::SWComponentTemplate::ApplicationAttributes			
Note	Annotation to a port regarding a certain VariableDataPrototype.			
Base	*ARObject*, *GeneralAnnotation*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
variable	VariableDataPrototype	1	ref	The instance of nv data annotated.

Table A.448: NvDataPortAnnotation

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*, *ServiceNeeds*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
testId	PositiveInteger	0..1	attr	Test Identifier (TID) according to ISO 15031-5.

Table A.449: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dataLength	PositiveInteger	0..1	attr	This attribute is applicable only if the ServiceNeeds is aggregated within BswModuleDependency. This attribute represents the length of data (in bytes) provided for this InfoType.
infoType	PositiveInteger	0..1	attr	The InfoType according to ISO 15031-5

Table A.450: 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 asssignedPort. 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dataLength	PositiveInteger	0..1	attr	This attribute is applicable only if the ServiceNeeds is aggregated within BswModuleDependency. This attribute represents the length of data (in bytes) provided for this particular PID signal.
parameterId	PositiveInteger	0..1	attr	Standardized parameter identifier (PID) according to the OBD standard specified in attribute "standard".
standard	String	0..1	attr	Annotates the standard according to which the PID is given, e.g. "ISO15031-5" or "SAE J1979 Rev May 2007".

Table A.451: 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).
Literal	***Description***
apiUse	The IUMPR service (of the DEM) uses an explicit API to connect to the component or module. **Tags:** atp.EnumerationValue=0
observer	The IUMPR service (of the Dem) uses no API but "observes" the associated diagnostic event. **Tags:** atp.EnumerationValue=1

Table A.452: 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*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
connectionType	ObdRatioConnection KindEnum	1	attr	Defines how the DEM is connected to the component or module to perform the IUMPR (In use monitor performance ratio) service.
denominator Group	DiagnosticDenominator ConditionEnum	0..1	attr	The denominator Dem shall use to compute the ratio.
iumprGroup	NameToken	0..1	attr	Defines the IUMPR (In use monitor performance ratio) Group of the SAE standard. Note that possible values are not predefined by an enumeration meta-type in order to make the meta-model independent of the details of the SAE standard.
rateBased MonitoredEvent	DiagnosticEventNeeds	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.
usedSecondary Fid	FunctionInhibitionNeeds	*	ref	This represents the secondary Function Inhibition Identifier used for the rate based monitor. This is an optional attribute. Any of the FID inhibitions leads to an inhibition of the IUMPR calculation

Table A.453: ObdRatioServiceNeeds

Class	OperationInvokedEvent			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents			
Note	The OperationInvokedEvent references the ClientServerOperation invoked by the client.			
Base	*ARObject*, *AbstractEvent*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *RTEEvent*, *Referrable*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
operation	ClientServerOperation	0..1	iref	The operation to be executed as the consequence of the event.

Table A.454: OperationInvokedEvent

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*			
Subclasses	ModeSwitchSenderComSpec, NvProvideComSpec, ParameterProvideComSpec, *SenderComSpec*, ServerComSpec			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
–	–	–	–	–

Table A.455: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
provided Interface	PortInterface	1	tref	The interface that this port provides. **Stereotypes:** isOfType

Table A.456: 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*, *Atp Feature*, *AtpPrototype*, *Identifiable*, *MultilanguageReferrable*, *PortPrototype*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
provided Required Interface	PortInterface	1	tref	This represents the PortInterface used to type the PRPort Prototype **Stereotypes:** isOfType

Table A.457: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.458: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
accessed Parameter	AutosarParameterRef	1	aggr	Refernce 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.

Table A.459: ParameterAccess

Class	**ParameterDataPrototype**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::DataPrototypes			
Note	A parameter element used for parameter interface and internal behavior, supporting signal like parameter and characteristic value communication patterns and parameter and characteristic value definition.			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *AutosarDataPrototype*, *DataPrototype*, *Identifiable*, *Multilanguage Referrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
initValue	ValueSpecification	0..1	aggr	Specifies initial value(s) of the ParameterDataPrototype

Table A.460: ParameterDataPrototype

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
parameter	ParameterData Prototype	1..*	aggr	The ParameterDataPrototype of this ParameterInterface.

Table A.461: ParameterInterface

Class	**ParameterPortAnnotation**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::ApplicationAttributes			
Note	Annotation to a port used for calibration regarding a certain ParameterDataPrototype.			
Base	*ARObject*, *GeneralAnnotation*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
parameter	ParameterData Prototype	1	ref	The instance of annotated ParameterDataPrototype.

Table A.462: ParameterPortAnnotation

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
constant Mapping	ConstantSpecification MappingSet	*	ref	Reference to the ConstanSpecificationMapping 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

△

Class	**ParameterSwComponentType**			
instantiation DataDefProps	InstantiationDataDefProps	*	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:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table A.463: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
providedOuterPort	AbstractProvidedPortPrototype	1	ref	This represents the provided outer delegation Port Prototype of the PassThroughSwConnector.
requiredOuterPort	AbstractRequiredPortPrototype	1	ref	This represents the required outer delegation Port Prototype of the PassThroughSwConnector.

Table A.464: 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*, *PackageableElement*, *Referrable*			
Subclasses	GeneralPurposePdu, *IPdu*, NmPdu, UserDefinedPdu			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
hasDynamicLength	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	Integer	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.465: Pdu

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*			
Attribute	***Type***	***Mul.***	***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.466: PduToFrameMapping

Class	**PduTriggering**
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication

▽

△

Class	**PduTriggering**			
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.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***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:** atpVariation **Tags:** 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.467: PduTriggering

Class	**PdurIPduGroup**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	The AUTOSAR PduR will enable and disable the sending of configurable groups of IPdus during runtime according to the AUTOSAR PduR specification. **Tags:** atp.recommendedPackage=PdurIPduGroups			
Base	*ARObject*, *CollectableElement*, *FibexElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

▽

△

Class	**PdurIPduGroup**			
communication Mode	String	1	attr	This attribute defines the use-case for this PduRIPdu Group. 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.
iPdu	PduTriggering	*	ref	Reference to a set of IPdus, which are contained in the PduR I-Pdu Group. If an IPdu is routed by the PduR to different destinations (PduR fan-out) than an Pdu Triggering for each destination is created in the System Template. To enable/disable a specific destination the PdurIPduGroup refers to the PduTriggering. atpVariation: The content of a PduR I-Pdu group can vary (->vehicle modes). **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild

Table A.468: PdurIPduGroup

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*			
Attribute	***Type***	***Mul.***	***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.
type	CIdentifier	1	attr	The name of the "C"-type
typeDefinition	String	1	attr	A definition of the type with the syntax of a 'C' typedef.

Table A.469: PerInstanceMemory

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
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

▽

△

Class	PhysConstrs			
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.470: 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			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
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:** atpVariation **Tags:** 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 must be variable, too. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=postBuild

▽

△

Class	***PhysicalChannel*** (abstract)			
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 must be variable, too. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, 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 must be variable, too. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=postBuild

Table A.471: 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*			
Attribute	***Type***	***Mul.***	***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

▽

△

Class	**PhysicalDimension**			
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.472: PhysicalDimension

Class	**PhysicalDimensionMapping**			
Package	M2::MSR::AsamHdo::Units			
Note	This class represents a specific mapping between two PhysicalDimensions.			
Base	*ARObject*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
firstPhysical Dimension	PhysicalDimension	1	ref	This represents the first PhysicalDimension of the enclosing PhysicalDimensionMapping.
secondPhysical Dimension	PhysicalDimension	1	ref	This represents the first PhysicalDimension of the enclosing PhysicalDimensionMapping.

Table A.473: PhysicalDimensionMapping

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
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.
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 conntected 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.
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.

▽

△

Class	**PncMapping**			
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.474: PncMapping

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*			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
enableTake Address	Boolean	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	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	1	ref	The option is valid for generated functions related to communication over this port
portArg Value (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.

Table A.475: 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*			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
value	ValueSpecification	1	aggr	Specifies the actual value.
valueType	ImplementationData Type	1	tref	The implementation type of this argument value. It should not be composite type or a pointer. **Stereotypes:** isOfType

Table A.476: PortDefinedArgumentValue

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*			
Attribute	***Type***	***Mul.***	***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.
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:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table A.477: 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			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
isService	Boolean	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.
serviceKind	ServiceProviderEnum	0..1	attr	This attribute provides further details about the nature of the applied service.

Table A.478: 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			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.479: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
clientServer Annotation	ClientServerAnnotation	*	aggr	Annotation of this PortPrototype with respect to client/server communication.
delegatedPort Annotation	DelegatedPort Annotation	0..1	aggr	Annotations on this delegated port.
ioHwAbstraction Server Annotation	IoHwAbstractionServer Annotation	*	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	ParameterPort Annotation	*	aggr	Annotations on this parameter port.
senderReceiver Annotation	SenderReceiver Annotation	*	aggr	Collection of annotations of this ports sender/receiver communication.
triggerPort Annotation	TriggerPortAnnotation	*	aggr	Annotations on this trigger port.

Table A.480: PortPrototype

Class	**PortPrototypeBlueprint**
Package	M2::AUTOSARTemplates::StandardizationTemplate::BlueprintDedicated::PortProtoypeBlueprint
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*

▽

△

Class	**PortPrototypeBlueprint**			
Attribute	***Type***	***Mul.***	***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.481: PortPrototypeBlueprint

Class	**PostBuildVariantCondition**			
Package	M2::AUTOSARTemplates::GenericStructure::VariantHandling			
Note	This class specifies the value which must 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*			
Attribute	***Type***	***Mul.***	***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.482: 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*			
Attribute	***Type***	***Mul.***	***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.483: PredefinedVariant

Class	**PrimitiveAttributeCondition**			
Package	M2::AUTOSARTemplates::StandardizationTemplate::DataExchangePoint::DataFormatTailoring			
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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
attribute	PrimitiveAttribute Tailoring	1	ref	The primitive attribute that has to be accepted by the restrictions of this PrimitiveAttributeCondition

Table A.484: PrimitiveAttributeCondition

Class	**PrimitiveAttributeTailoring**			
Package	M2::AUTOSARTemplates::StandardizationTemplate::DataExchangePoint::DataFormatTailoring			
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*			
Attribute	***Type***	***Mul.***	***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.485: PrimitiveAttributeTailoring

Class	**ProvidedServiceInstance**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Service instances that are provided by the ECU that is connected via the ApplicationEndpoint to a CommunicationConnector.			
Base	*ARObject*, *AbstractServiceInstance*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
EventHandler	EventHandler	*	aggr	Collection of event callback configurations.
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.
priority	PositiveInteger	0..1	attr	Defines the frame priority where values from 0 (best effort) to 7 (highest) are allowed.
sdServerConfig	SdServerConfig	0..1	aggr	Service Discovery Server configuration.
serviceIdentifier	PositiveInteger	0..1	attr	Service ID. Shall be unique within one system to allow service discovery.

Table A.486: ProvidedServiceInstance

Class	**QueuedReceiverComSpec**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	Communication attributes specific to queued receiving.			
Base	*ARObject*, *RPortComSpec*, *ReceiverComSpec*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
queueLength	PositiveInteger	1	attr	Length of queue for received events.

Table A.487: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.488: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.489: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
required Interface	PortInterface	1	tref	The interface that this port requires, i.e. the port depends on another port providing the specified interface. **Stereotypes:** isOfType

Table A.490: 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*, *MultilanguageReferrable*, *Referrable*			
Subclasses	AsynchronousServerCallReturnsEvent, BackgroundEvent, DataReceiveErrorEvent, DataReceivedEvent, DataSendCompletedEvent, DataWriteCompletedEvent, ExternalTriggerOccurredEvent, InitEvent, InternalTriggerOccurredEvent, ModeSwitchedAckEvent, OperationInvokedEvent, SwcModeManagerErrorEvent, SwcModeSwitchEvent, TimingEvent, TransformerHardErrorEvent			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
disabledMode	ModeDeclaration	*	iref	Reference to the Modes that disable the Event. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=contextPort, contextModeDeclarationGroupPrototype, targetModeDeclaration
startOnEvent	RunnableEntity	0..1	ref	RunnableEntity starts when the corresponding RTEEvent occurs.

Table A.491: 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*, *PackageableElement*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
hostSystem	System	1	ref	System which describes the software components of the host ECU.
rptContainer	RptContainer	1..*	aggr	Top-level rptContainer definitions of this specific rapid prototyping scenario. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabelvh.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=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.492: RapidPrototypingScenario

Class	***ReceiverComSpec*** (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	Receiver-specific communication attributes (RPortPrototype typed by SenderReceiverInterface).			
Base	*ARObject*, *RPortComSpec*			
Subclasses	NonqueuedReceiverComSpec, QueuedReceiverComSpec			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
compositeNetworkRepresentation	CompositeNetworkRepresentation	*	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.
handleOutOfRange	HandleOutOfRangeEnum	1	attr	This attribute controls how values that are out of the specified range are handled according to the values of HandleOutOfRangeEnum.
handleOutOfRangeStatus	HandleOutOfRangeStatusEnum	0..1	attr	Control the way how return values are created in case of an out-of-range situation.
maxDeltaCounterInit	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. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
maxNoNewOrRepeatedData	PositiveInteger	0..1	attr	The maximum amount of missing or repeated Data which the receiver does not expect to exceed under normal communication conditions.
networkRepresentation	SwDataDefProps	0..1	aggr	A networkRepresentation is used to define how the data Element is mapped to a communication bus.
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 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.
transformationComSpecProps	TransformationComSpecProps	*	aggr	This references the TransformationComSpecProps which define port-specific configuration for data transformation.
usesEndToEndProtection	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.493: ReceiverComSpec

Class	**RecordValueSpecification**			
Package	M2::AUTOSARTemplates::CommonStructure::Constants			
Note	Specifies the values for a record.			
Base	*ARObject*, *CompositeValueSpecification*, *ValueSpecification*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
field (ordered)	ValueSpecification	1..*	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 indepenently of the shortNames. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table A.494: RecordValueSpecification

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	***Datatype***	***Mul.***	***Kind***	***Note***
base	Identifier	0..1	attr	This attribute reflects the base to be used for this reference. **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:** 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. **Tags:** xml.attribute=true

Table A.495: 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 must be unique in the current package.
Base	*ARObject*

▽

△

Class	**ReferenceBase**			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
baseIsThis Package	Boolean	1	attr	This indicates that this base is established by the current package. In this case the association "package" can be derived as the qualified shortName of the enclosing package. If the value of baseIsThisPackage is set to true then one of the following must be true: • target of the association "package" must be the enclosing package. • association "package" is omitted. **Tags:** xml.sequenceOffset=28
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:** xml.sequenceOffset=20
isGlobal	Boolean	1	attr	This indicates that the target of the applicable reference can be resolved via the non-qualified shortName. This requires that the shortName of the target is unique within the package referenced in the reference base. The default is false. Note that the reference base also maintains a list of elements which may be referenced using a "global Reference". **Tags:** xml.sequenceOffset=25
package	ARPackage	0..1	ref	This association specifies the basis of all relative references with the base equals shortLabel. This association must exist unless the value of baseIsThis Package is set to true. **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. **Tags:** xml.sequenceOffset=10

Table A.496: ReferenceBase

Class	**ReferenceCondition**			
Package	M2::AUTOSARTemplates::StandardizationTemplate::DataExchangePoint::DataFormatTailoring			
Note	The ReferenceCondition evaluates to true, if the referenced reference is accepted by all rules of this condition.			
Base	*ARObject*, *AbstractCondition*, *AbstractMultiplicityRestriction*, *AttributeCondition*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
reference	ReferenceTailoring	1	ref	The reference that has to be accepted by the restrictions of this ReferenceCondition

Table A.497: ReferenceCondition

Class	**ReferenceTailoring**			
Package	M2::AUTOSARTemplates::StandardizationTemplate::DataExchangePoint::DataFormatTailoring			
Note	Tailoring of Non-Containment References.			
Base	*ARObject*, *AttributeTailoring*, *DataFormatElementReference*, *DataFormatElementScope*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *SpecElementReference*, *SpecElementScope*			
Attribute	***Type***	***Mul.***	***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.498: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
referenceValue	DataPrototype	1	ref	The referenced data prototype.

Table A.499: 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, BswVariableAccess, CouplingPortTrafficClassAssignment, DiagnosticDebounceAlgorithmProps, *DiagnosticEnvModeElement*, EthernetPriorityRegeneration, EventHandler, ExclusiveAreaNestingOrder, *HwDescriptionEntity*, *ImplementationProps*, LinSlaveConfigIdent, ModeTransition, *MultilanguageReferrable*, PncMappingIdent, *SingleLanguageReferrable*, SocketConnectionBundle, TimeSyncServerConfiguration, TpConnectionIdent			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

▽

△

Class	***Referrable*** (abstract)			
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. **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.500: Referrable

Class	**RoleBasedDataAssignment**			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	This class specifies an assignment of a role to a particular data object in the SwcInternalBehavior 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 can be mapped to a specific ServiceNeeds element, so that a tool is able to create the correct access.			
Base	*ARObject*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
role	Identifier	1	attr	This is the role of the assigned data in the given context, for example for an NVRAM Block it is used to distinguish between an mirror block and a ROM default block. Possible values need to be specified on M1 level. This also is intended to support the so called "Signal based Approach" of the DCM. In this use case the name of the involved data element is required. This name shall be taken from the DataElement referenced by the property usedDataElement. The following values are standardized: • **ramBlock** indicates data to be used as a mirror for an NVRAM Block. • **defaultValue** indicates constant data to be used as default in the context of this ServiceNeeds, e.g. for an NVRAM Block. • **signalBasedDiagnostics** indicates the Role BasedDataAssignment shall be used for signal based diagnostics.
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.

▽

△

Class	**RoleBasedDataAssignment**			
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.501: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
role	Identifier	1	attr	This is the role of the associated data type in the given context.
used Implementation DataType	ImplementationData Type	1	ref	This represents the associated ImplementationDataType.

Table A.502: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
portPrototype	PortPrototype	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	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.503: RoleBasedPortAssignment

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
category	CalprmAxisCategory Enum	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	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	1	aggr	For multidimensional compound primitivies (curve, map ...) it is necessary to know the dimensions.They are specified using swArraySize. **Tags:** xml.sequenceOffset=40
swAxisIndex	AxisIndexType	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.504: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
ruleBased Values	RuleBasedValue Specification	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 primitivies CURVE, MAP, CUBOID, CUBE_4, CUBE_5, COM_AXIS, RES_AXIS, VAL_BLK, STRING. 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

△

Class	**RuleBasedValueCont**			
unit	Unit	0..1	ref	This represents the physical unit of the provided values. **Tags:** xml.sequenceOffset=30

Table A.505: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
arguments	RuleArguments	1	aggr	This represents the arguments for the RuleBasedValue Specification. **Stereotypes:** atpVariation **Tags:** 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	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.506: 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*			
Attribute	***Type***	***Mul.***	***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=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
canBeInvoked Concurrently	Boolean	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. Note that the default value of this attribute is set to "false".

▽

△

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=shortName, 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=shortName, 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 dataReceivePointByValue is subject to variability with the purpose to support the conditional existence of sender receiver ports or the variant existence of data receive points in the implementation. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, 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=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
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 ▽

▽

△

Class	**RunnableEntity**			
				△ existence of sender receiver ports or the variant existence of dataWriteAccess in the implementation. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, 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, variation Point.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=shortName, 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, variation Point.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=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
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=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	**RunnableEntity**			
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=shortName, 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=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
symbol	CIdentifier	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=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table A.507: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
symbol	CIdentifier	1	attr	This represents the symbol to be generated into the actual signature on the level of the C programming language.

Table A.508: RunnableEntityArgument

Class	**RunnableEntityGroup**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::ImplicitCommunicationBehavior			
Note	This meta-class represents the ability to define a collection of RunnableEntities. The collection can be nested.			
Base	*ARObject*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
runnableEntity	RunnableEntity	*	iref	This represents a collection of RunnableEntitys that belong to the enclosing RunnableEntityGroup. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
runnableEntity Group	RunnableEntityGroup	*	iref	This represents the ability to define nested groups of RunnableEntitys. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table A.509: RunnableEntityGroup

Enumeration	**RxAcceptContainedIPduEnum**
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication
Note	Defines whether this ContainerIPdu has a fixed set of containedIPdus assigned for reception.
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.EnumerationValue=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.EnumerationValue=1

Table A.510: RxAcceptContainedIPduEnum

Class	**SOMEIPTransformationDescription**			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	The SOMEIPTransformationDescription is used to specify SOME/IP transformer specific attributes.			
Base	*ARObject*, *Describable*, *TransformationDescription*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
alignment	PositiveInteger	1	attr	Specifies the alignment of dynamic data in the serialized data stream. 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.

Table A.511: SOMEIPTransformationDescription

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*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
implements SOMEIPString Handling	Boolean	0..1	attr	This attribute indicates whether Strings in the SOME/IP message shall be processed according to the SOME/IP specification for Strings. This attribute has been introduced due to compatibility reasons for AUTOSAR before R4.3. If this attribute is set to true Strings in the payload shall be handled according to the SOME/IP specification on Strings. If this attribute is set to false (or not set) no special handling for Strings in the payload shall be performed.
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 in the SOME/IP message. This attribute is valid for all available occurrences of fixed-size arrays in the SOME/IP message. For a more fine granular modeling on the level of DataPrototypes the DataPrototypeTransformationProps shall be used.
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	TlvDataIdDefinition	*	aggr	This aggregation represents the collection of tlvDataIds defined in the enclosing context. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=tlvDataId atp.Status=draft

Table A.512: SOMEIPTransformationISignalProps

Class	**SOMEIPTransformationProps**			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	The class SOMEIPTransformationProps specifies SOME/IP specific configuration properties.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TransformationProps*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**

▽

△

Class	**SOMEIPTransformationProps**			
alignment	PositiveInteger	0..1	attr	Specifies the alignment of dynamic data in the serialized data stream. The alignment is 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 a static size Array 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.513: SOMEIPTransformationProps

Class	**SdClientConfig**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Client configuration for Service-Discovery.			
Base	*ARObject*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
capability Record	TagWithOptionalValue	*	aggr	A sequence of records to store arbitrary name/value pairs conveying additional information about the named service. Capability records shall only be existing if the respective SdClientConfig is composed by a ConsumedService Instance (see constr_3260).
clientService MajorVersion	PositiveInteger	0..1	attr	Major version number of the Service.
clientService MinorVersion	PositiveInteger	0..1	attr	Minor version number of the Service.
initialFind Behavior	InitialSdDelayConfig	0..1	aggr	Controls initial find behavior of clients.
request ResponseDelay	RequestResponseDelay	0..1	aggr	Maximum/Minimum allowable response delay to entries received by multicast in seconds.
ttl	PositiveInteger	1	attr	TTL for Request and Subscribe messages.

Table A.514: SdClientConfig

Class	**SdServerConfig**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Server configuration for Service-Discovery.			
Base	*ARObject*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
capability Record	TagWithOptionalValue	*	aggr	A sequence of records to store arbitrary name/value pairs conveying additional information about the named service. Capability records shall only be existing if the respective SdServerConfig is composed by a ProvidedService Instance (see constr_3259).
initialOffer Behavior	InitialSdDelayConfig	0..1	aggr	Controls offer behavior of the server.

▽

△

Class	**SdServerConfig**			
offerCyclicDelay	TimeValue	0..1	attr	Optional attribute to define cyclic offers. Cyclic offer is active, if the delay is set (in seconds).
request ResponseDelay	RequestResponseDelay	0..1	aggr	Maximum/Minimum allowable response delay to entries received by multicast in seconds.
serverService MajorVersion	PositiveInteger	0..1	attr	Major version number of the Service.
serverService MinorVersion	PositiveInteger	0..1	attr	Minor version number of the Service.
ttl	PositiveInteger	1	attr	Time to live. Shall be a positive value (sInt32).

Table A.515: SdServerConfig

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*			
Attribute	***Type***	***Mul.***	***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.516: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
sdgClass	SdgClass	*	aggr	The owned sdgClasses which define the structure of the Sdgs **Tags:** xml.namePlural=SDG-CLASSES

Table A.517: SdgDef

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*			
Attribute	***Type***	***Mul.***	***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.518: SectionNamePrefix

Class	**SecureCommunicationAuthenticationProps**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Authentication properties used to configure SecuredIPdus.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
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.519: SecureCommunicationAuthenticationProps

Class	**SecureCommunicationFreshnessProps**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Freshness properties used to configure SecuredIPdus.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***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.

△

Class	**SecureCommunicationFreshnessProps**			
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.520: SecureCommunicationFreshnessProps

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
authentication Props	SecureCommunication AuthenticationProps	0..1	ref	Reference to authentication properties that are valid for this SecuredIPdu.
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.521: SecuredIPdu

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*			
Attribute	***Type***	***Mul.***	***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.522: 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			
Attribute	***Type***	***Mul.***	***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	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.
transmission Acknowledge	Transmission Acknowledgement Request	0..1	aggr	Requested transmission acknowledgement for data element.
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.523: 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 SenderReceiverToSignalGroupMapping is typed by an ApplicationDataType the reference to the ApplicationArrayElement 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*			
Attribute	***Type***	***Mul.***	***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.524: 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			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.525: 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.

▽

△

Class	**SenderRecRecordElementMapping**			
Base	*ARObject*			
Attribute	***Type***	***Mul.***	***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.
systemSignal	SystemSignal	0..1	ref	Reference to the system signal used to carry the primitive ApplicationRecordElement.

Table A.526: 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			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
computed	Boolean	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	1	ref	The instance of VariableDataPrototype annotated.
limitKind	DataLimitKindEnum	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 AtomicSwComponentType is transmitted to another AtomicSwComponentType 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	1	attr	This attribute controls how data is processed according to the possible values of ProcessingKindEnum.

Table A.527: 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*			
Attribute	***Type***	***Mul.***	***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.
systemSignal	SystemSignal	1	ref	Reference to the SystemSignal to which one primitive of the composite type is mapped.

▽

△

Class	**SenderReceiverCompositeElementToSignalMapping**			
typeMapping	SenderRecComposite TypeMapping	1	aggr	The CompositeTypeMapping maps one VariableData Prototype of the composite data type to a SystemSignal.

Table A.528: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dataElement	VariableDataPrototype	1..*	aggr	The data elements of this SenderReceiverInterface.
invalidation Policy	InvalidationPolicy	*	aggr	InvalidationPolicy for a particular dataElement

Table A.529: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dataElement	VariableDataPrototype	1	iref	Reference to a data element with a composite datatype which is mapped to a signal group.
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 the Application ArrayElements and ApplicationRecordElements to Signals of the SignalGroup.

Table A.530: SenderReceiverToSignalGroupMapping

Class	**SenderReceiverToSignalMapping**			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	Mapping of a sender receiver communication data element to a signal.			
Base	*ARObject*, *DataMapping*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dataElement	VariableDataPrototype	1	iref	Reference to the data element.
systemSignal	SystemSignal	1	ref	Reference to the system signal used to carry the data element.

Table A.531: 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*, *AtpType*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *SwComponentType*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
sensorActuator	HwDescriptionEntity	1	ref	Reference from the Sensor Actuator Software Component Type to the description of the actual hardware.

Table A.532: SensorActuatorSwComponentType

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			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
operation	ClientServerOperation	0..1	iref	The operation that is called by this runnable.
timeout	TimeValue	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.533: ServerCallPoint

Class	**ServerComSpec**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	Communication attributes for a server port (PPortPrototype and ClientServerInterface).			
Base	*ARObject*, *PPortComSpec*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
operation	ClientServerOperation	0..1	ref	Operation these communication attributes apply to.
queueLength	PositiveInteger	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.534: ServerComSpec

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*			
Subclasses	BswMgrNeeds, ComMgrUserNeeds, CryptoServiceJobNeeds, CryptoServiceNeeds, *Diagnostic CapabilityElement*, DltUserNeeds, *DoIpServiceNeeds*, EcuStateMgrUserNeeds, ErrorTracerNeeds, FunctionInhibitionAvailabilityNeeds, FunctionInhibitionNeeds, GlobalSupervisionNeeds, HardwareTest Needs, IndicatorStatusNeeds, J1939RmIncomingRequestServiceNeeds, J1939RmOutgoingRequest ServiceNeeds, NvBlockNeeds, SecureOnBoardCommunicationNeeds, SupervisedEntityCheckpoint Needs, SupervisedEntityNeeds, SyncTimeBaseMgrUserNeeds, V2xFacUserNeeds, V2xMUserNeeds, VendorSpecificServiceNeeds			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.535: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.536: 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*			

▽

△

Class	**ServiceSwComponentType**			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.537: ServiceSwComponentType

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
maximum ExecutionTime	MultidimensionalTime	1	aggr	The maximum simulated execution time.
minimum ExecutionTime	MultidimensionalTime	1	aggr	The minimum simulated execution time.
nominal ExecutionTime	MultidimensionalTime	1	aggr	The nominal simulated execution time.

Table A.538: SimulatedExecutionTime

Class	**SoAdConfig**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication			
Note	SoAd Configuration for one specific Physical Channel.			
Base	*ARObject*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
connection Bundle	SocketConnection Bundle	*	aggr	Collection of SocketConnectionBundles. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
socketAddress	SocketAddress	1..*	aggr	Collection of SoAdAddresses. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild

Table A.539: SoAdConfig

Class	**SocketAddress**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication			
Note	This meta-class represents the ability represent a socket address towards the rest of the meta-model. The actual semantics of the represented socket address, however, is contributed by aggregation of ApplicationEndpoint.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
application Endpoint	ApplicationEndpoint	0..1	aggr	Application addressing

▽

△

Class	**SocketAddress**			
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. In a System Description this reference is mandatory if an IP unicast address is described. In an ECU Extract this reference is optional (references to ECUs that are not part of the ECU Extract shall be avoided). Please note that in the SystemExtract or EcuExtract the type of the reference shall not change from unicast connectorRef to multicastConnectorRef.
multicast Connector	EthernetCommunication Connector	*	ref	Association to a CommunicationConnector in the topology description. This reference shall be used if the SocketAddress describes an IP mutlicast address. This multicast SocketAddress shall contain references to all ECUs that want to receive the multicast messages. In a System Description this reference is mandatory if an IP multicast address is described. In an ECU Extract this reference is optional (references to ECUs that are not part of the ECU Extract shall be avoided). Please note that in the SystemExtract or EcuExtract the type of the reference shall not change from multicastConnectorRef to unicast connectorRef.

Table A.540: SocketAddress

Class	**SocketConnection**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication			
Note	The SoAd serves as a (De)Multiplexer between different PDU sources and the TCP/IP stack.			
Base	*ARObject*, *Describable*			
Attribute	***Type***	***Mul.***	***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.
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.
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.
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.

▽

△

Class	**SocketConnection**			
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.541: SocketConnection

Class	**SocketConnectionBundle**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication			
Note	This elements groups SocketConnections, i.e. specifies socket connections belonging to the bundle and describes properties which are common for all socket connections in the bundle.			
Base	*ARObject*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
bundled Connection	SocketConnection	1..*	aggr	Collection of SocketConnections in the connectionGroup.
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.
pathMtu Discovery Enabled	Boolean	0..1	attr	Defines whether the Path MTU Discovery shall be performed for the related socket.
pdu	SocketConnectionIpdu Identifier	*	aggr	With this aggregation SocketConnectionIpduIdentifier elements are assigned to all SocketConnections that are available in this SocketConnetionBundle.
serverPort	SocketAddress	1	ref	Server Port for TCP/UDP connection in an abstract communication sense. The server is the major provider of the communication. Please note that the server may also consume data.
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.542: SocketConnectionBundle

Class	**SocketConnectionIpduIdentifier**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication			
Note	An Identifier is required in case of one port per ECU communication where multiple Pdus are transmitted over the same connection. If only one IPdu is transmitted over the connetion this attribute can be ignored.			
Base	*ARObject*			
Attribute	***Type***	***Mul.***	***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.
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 mapped to a socket connection.
routingGroup	SoAdRoutingGroup	*	ref	Reference to RoutingGroups that can be enabled or disabled.

Table A.543: SocketConnectionIpduIdentifier

Class	**SomeipTpConnection**			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	A connection identifies the sender and the receiver of this particular communication. The SOME/IP TP module routes a Pdu through this connection.			
Base	*ARObject*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
tpChannel	SomeipTpChannel	0..1	ref	Assignment of configuration properties valid for this SomeipTpConnection.
tpSdu	PduTriggering	0..1	ref	Reference to an IPdu that is segmented by the Transport Protocol.
transportPdu	PduTriggering	0..1	ref	Reference to the segmented IPdu.

Table A.544: SomeipTpConnection

Class	***SpecElementReference*** (abstract)			
Package	M2::AUTOSARTemplates::StandardizationTemplate::DataExchangePoint::CommonPatterns			
Note	This is a reference to a specification element in the Autosar standard.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Subclasses	*DataFormatElementReference*, *SpecElementScope*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
alternative Name	String	0..1	attr	Alternative name of a specification element if its name doesn't fit into the shortName. E.g. because the name contains spaces.

Table A.545: SpecElementReference

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*			
Attribute	*Type*	*Mul.*	*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.546: StaticPart

Class	**Std**			
Package	M2::MSR::Documentation::TextModel::InlineTextElements			
Note	This represents a reference to external standards.			
Base	*ARObject*, *Referrable*, *SingleLanguageReferrable*			
Attribute	*Type*	*Mul.*	*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.547: 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*			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
firstElement	SubElementRef	0..1	aggr	This represents the first element referenced in the scope of the mapping. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
secondElement	SubElementRef	0..1	aggr	This represents the second element referenced in the scope of the mapping. **Stereotypes:** atpVariation **Tags:** 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.548: SubElementMapping

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			
Base	*ARElement*, *ARObject*, *AtpBlueprint*, *AtpBlueprintable*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Attribute	***Type***	***Mul.***	***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 addresssing method.

Table A.549: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
category	CalprmAxisCategory Enum	1	attr	This category specifies the particular axis types: • STD_AXIS • COM_AXIS • RES_AXIS (swArraysize necessary) **Tags:** xml.sequenceOffset=20

▽

△

Class	**SwAxisCont**			
swArraysize	ValueList	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
swAxisIndex	AxisIndexType	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	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.550: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
swAxisType	SwAxisType	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.551: SwAxisGeneric

Class	**SwAxisGrouped**			
Package	M2::MSR::DataDictionary::Axis			
Note	An SwAxisGrouped is an axis which is shared between multiple calibration parameters.			
Base	*ARObject*, *SwCalprmAxisTypeProps*			
Attribute	***Type***	***Mul.***	***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.552: 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*			
Attribute	***Type***	***Mul.***	***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	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	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
swVariable Ref (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.553: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.554: SwBaseType

Enumeration	**SwCalibrationAccessEnum**
Package	M2::MSR::DataDictionary::DataDefProperties
Note	Determines the access rights to a data object w.r.t. measurement and calibration.
Literal	***Description***
notAccessible	The element will not be accessible via MCD tools, i.e. will not appear in the ASAP file. **Tags:** atp.EnumerationValue=0
readOnly	The element will only appear as read-only in an ASAP file. **Tags:** atp.EnumerationValue=1
readWrite	The element will appear in the ASAP file with both read and write access. **Tags:** atp.EnumerationValue=2

Table A.555: SwCalibrationAccessEnum

Class	**SwComponentPrototype**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Composition			
Note	Role of a software component within a composition.			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
type	SwComponentType	1	tref	Type of the instance. **Stereotypes:** isOfType

Table A.556: SwComponentPrototype

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			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
consistency Needs	ConsistencyNeeds	*	aggr	This represents the collection of ConsistencyNeeds owned by the enclosing SwComponentType. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, 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=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
portGroup	PortGroup	*	aggr	A port group being part of this component. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

▽

△

Class	***SwComponentType*** (abstract)			
swComponent Documentation	SwComponent Documentation	0..1	aggr	This adds a documentation to the SwComponentType. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=swComponentDocumentation, 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.557: 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			
Attribute	***Type***	***Mul.***	***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.558: 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*

▽

△

Class	«atpVariation» **SwDataDefProps**			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
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 alignment of the DataPrototype. If the attribute is not defined the alignment is determined by the swBaseType size and the memory AllocationKeywordPolicy of the referenced SwAddr Method. **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

▽

△

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 Size Mult (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

▽

△

Class	«atpVariation» **SwDataDefProps**			
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.559: SwDataDefProps

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*			
Attribute	***Type***	***Mul.***	***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. **Tags:** 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. Since currently no categories for BswModuleEntry are defined it will be empty. **Tags:** xml.sequenceOffset=5

Table A.560: SwPointerTargetProps

Class	**SwRecordLayout**			
Package	M2::MSR::DataDictionary::RecordLayout			
Note	Defines how the data objects (variables, calibration parameters etc.) are to be stored in the ECU memory. As an example, this definition specifies the sequence of axis points in the ECU memory. Iterations through axis values are stored within the sub-elements swRecordLayoutGroup. **Tags:** atp.recommendedPackage=SwRecordLayouts			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

▽

△

Class	**SwRecordLayout**			
swRecord LayoutGroup	SwRecordLayoutGroup	1	aggr	This is the top level record layout group. **Tags:** xml.roleElement=true xml.roleWrapperElement=false xml.sequenceOffset=20 xml.typeElement=false xml.typeWrapperElement=false

Table A.561: SwRecordLayout

Class	**SwRecordLayoutV**			
Package	M2::MSR::DataDictionary::RecordLayout			
Note	This element specifies which values are stored for the current SwRecordLayoutGroup. If no baseType is present, the SwBaseType referenced initially in the parent SwRecordLayoutGroup is valid. The specification of swRecordLayoutVAxis gives the axis of the values which shall be stored in accordance with the current record layout SwRecordLayoutGroup. In swRecordLayoutVProp one can specify the information which shall be stored.			
Base	*ARObject*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
desc	MultiLanguageOverview Paragraph	0..1	aggr	This aggregation allows for a brief description about the particular record layout value which can help to identify the entry. In-depth documentation should be added to the introduction of the surrounding record layout. **Tags:** xml.sequenceOffset=20
category	AsamRecordLayout Semantics	0..1	attr	This attribute denotes the semantics in particular in terms of the corresponding A2L-Keyword. This is to support the mapping of the more general record layouts in AUTOSAR/MSR to the specific A2l keywords. It is possible to express the specific semantics of A2l Record Layout keywords in swRecordlayoutGroup but not always vice versa. Therefore the mapping is provided in this optional attribute. **Tags:** xml.sequenceOffset=5
baseType	SwBaseType	0..1	ref	This association allows to refer to a base type in case a specific encoding is intended. If no base type is referred, the base type referenced initially in the corresponding DataPrototype is to be used. **Tags:** xml.sequenceOffset=30
shortLabel	Identifier	1	attr	This attribute specifies a name which can be used e.g. when ECU code is generated from the record layout value. **Tags:** xml.sequenceOffset=3
swGenericAxis ParamType	SwGenericAxisParam Type	0..1	ref	This association supports the case that a value from a generic axis definition shall be stored. This value is denoted by a particular generic axis parameter type. **Tags:** xml.sequenceOffset=70

▽

△

Class	**SwRecordLayoutV**			
swRecord LayoutVAxis	AxisIndexType	0..1	attr	This attribute gives the index of the axis of which values that are stored in the record. swRecordVIndex refers to the symbolic names of the iterators for which the axis value shall be stored in the record. In case of nested iterators (mainly for multidimensional objects) the iterator names are specified as whitespace-separated names. These symbolic names relate to swRecordLayoutGroup Index. The iterators are processed from left to right in such a manner that they symbolize the loop index from the outside to the inside. It is considered an error if more components are specified than axes exist in the related ApplicationDataType. **Tags:** xml.sequenceOffset=40
swRecord LayoutVFix Value	Integer	0..1	attr	This attribute specifies the filler character for the current record layout, in the form of hex digits. It is also used to specify the fix value for e.g. FIXRIGHTDIFF. **Tags:** xml.sequenceOffset=80
swRecord LayoutVIndex	NameTokens	0..1	attr	The symbolic value for iteration, or the symbolic values separated by whitespaces, refer to the symbolic values given in swRecordLayoutGroupIndex . The iterators are processed from left to right, in such a manner that they symbolize the loop index from the outside to the inside. It is considered an error if the record layout is referenced by an entity which has less number of axes than index names referenced here. **Tags:** xml.sequenceOffset=60
swRecord LayoutVProp	NameToken	0..1	attr	This attribute describes the kind of values to be stored. More details see below. The standardized values foreseen for this attribute are defined in [TPS_SWCT_01489]. **Tags:** xml.sequenceOffset=50

Table A.562: SwRecordLayoutV

Class	**SwServiceArg**			
Package	M2::MSR::DataDictionary::ServiceProcessTask			
Note	Specifies the properties of a data object exchanged during the call of an SwService, e.g. an argument or a return value. The SwServiceArg can also be used in the argument list of a C-macro. For this purpose the category shall be set to "MACRO". A reference to implementationDataType can optional be added if the actual argument has an implementationDataType.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

▽

△

Class	**SwServiceArg**			
direction	ArgumentDirection Enum	0..1	attr	Specifies the direction of the data transfer. The direction shall indicate the direction of the actual information that is being consumed by the caller and/or the callee, not the direction of formal arguments in C. The attribute is optional for backwards compatibility reasons. For example, if a pointer is used to pass a memory address for the expected result, the direction shall be "out". If a pointer is used to pass a memory address with content to be read by the callee, its direction shall be "in". **Tags:** xml.sequenceOffset=10
swArraysize	ValueList	0..1	aggr	This turns the argument of the service to an array. **Tags:** xml.sequenceOffset=20
swDataDef Props	SwDataDefProps	0..1	aggr	Data properties of this SwServiceArg. **Tags:** xml.sequenceOffset=30

Table A.563: SwServiceArg

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
swDataDef Props	SwDataDefProps	0..1	aggr	This denotes the data defintion 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. **Tags:** xml.sequenceOffset=40

Table A.564: SwSystemconst

Class	**SwSystemconstValue**			
Package	M2::AUTOSARTemplates::GenericStructure::VariantHandling			
Note	This meta-class assigns a particular value to a system constant.			
Base	*ARObject*			
Attribute	***Type***	***Mul.***	***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

▽

△

Class	**SwSystemconstValue**			
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.565: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
sw Systemconstant Value	SwSystemconstValue	*	aggr	This is one particular value of a system constant.

Table A.566: SwSystemconstantValueSet

Class	**SwValueCont**			
Package	M2::MSR::CalibrationData::CalibrationValue			
Note	This metaclass represents the content of one particular SwInstance.			
Base	*ARObject*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
swArraysize	ValueList	0..1	aggr	This attribute defines the size of each dimension for compound primitivies CURVE, MAP, CUBOID, CUBE_4, CUBE_5, COM_AXIS, RES_AXIS, VAL_BLK, STRING. 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.567: SwValueCont

Class	**SwVariableRefProxy**			
Package	M2::MSR::DataDictionary::DatadictionaryProxies			
Note	Proxy class for several kinds of references to a variable.			
Base	*ARObject*			
Attribute	***Type***	***Mul.***	***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
mcDataInstance Var	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.568: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
bswBehavior	BswInternalBehavior	1	ref	The mapped BswInternalBehavior
runnable Mapping	SwcBswRunnable Mapping	*	aggr	A mapping between a pair of SWC and BSW runnables. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
swcBehavior	SwcInternalBehavior	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:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
synchronized Trigger	SwcBswSynchronized Trigger	*	aggr	A pair of SWC and BSW Triggers to be synchronized by the scheduler. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table A.569: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

▽

△

Class	**SwcBswRunnableMapping**			
bswEntity	BswModuleEntity	1	ref	The mapped BswModuleEntity
swcRunnable	RunnableEntity	1	ref	The mapped SWC runnable.

Table A.570: 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*			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
bswModeGroup	ModeDeclarationGroup Prototype	1	ref	The BSW mode group prototype.
swcModeGroup	ModeDeclarationGroup Prototype	1	iref	The SWC mode group prototype provided by a particular port.

Table A.571: 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*			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
bswTrigger	Trigger	1	ref	The BSW Trigger.
swcTrigger	Trigger	1	iref	The SWC Trigger provided by a particular port.

Table A.572: 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*			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
apiPrinciple	ApiPrincipleEnum	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	1	ref	This reference represents the ExclusiveArea for which the policy applies.

Table A.573: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
behavior	SwcInternalBehavior	1	ref	The internal behavior implemented by this Implementation.
perInstance MemorySize	PerInstanceMemory Size	*	aggr	Allows a definition of the size of the per-instance memory for this implementation. The aggregation of PerInstanceMemorySize 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 PerInstance Memory. **Stereotypes:** atpVariation **Tags:** 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.574: 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*			
Attribute	***Type***	***Mul.***	***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=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
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=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime

△

Class	**SwcInternalBehavior**			
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 vh.latestBindingTime=preCompileTime
explicitInter Runnable Variable	VariableDataPrototype	*	aggr	Implement state message semantics for establishing communication among runnables of the same component. The aggregation of explicitInterRunnableVariable 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=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
handle TerminationAnd Restart	HandleTerminationAnd RestartEnum	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 implicitInterRunnableVariable 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=shortName, 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
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 PortPrototypes and component local memories like "perInstanceParameter" or "arTypedPer InstanceMemory". **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=instantiationDataDefProps, variation Point.shortLabel vh.latestBindingTime=preCompileTime

▽

△

Class	SwcInternalBehavior			
perInstance Memory	PerInstanceMemory	*	aggr	Defines a per-instance memory object needed by this software component. The aggregation of PerInstanceMemory 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=shortName, 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=shortName, 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, variationPoint.short Label vh.latestBindingTime=preCompileTime
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=shortName, variationPoint.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 ▽

▽

△

Class	SwcInternalBehavior			
				△ for Obd related Service Needs) tools. Therefore the aggregation is «atpSplitable». **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, 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=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
supports Multiple Instantiation	Boolean	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=shortName

Table A.575: SwcInternalBehavior

Class	SwcModeSwitchEvent			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents			
Note	This event is raised upon a received mode change.			
Base	*ARObject*, *AbstractEvent*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *RTEEvent*, *Referrable*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
activation	ModeActivationKind	1	attr	Specifies if the event is activated on entering or exiting the referenced Mode.
mode (ordered)	ModeDeclaration	1..2	iref	Reference to one or two Modes that initiate the SwcMode SwitchEvent.

Table A.576: 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*			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
assignedData	RoleBasedData Assignment	*	aggr	Defines the role of an associated data object of the same component. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

▽

△

Class	**SwcServiceDependency**			
assignedPort	RoleBasedPort Assignment	*	aggr	Defines the role of an associated port of the same component. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=assignedPort, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
representedPortGroup	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	1	aggr	The associated ServiceNeeds.

Table A.577: 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*			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
applicationPartition	ApplicationPartition	0..1	ref	Reference to an ApplicationPartition to which a SwComponentPrototype is mapped.
swComponentPrototype	SwComponentPrototype	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 SwComponentPrototype INSIDE the Composition to another ApplicationPartition the inner mapping overrides the outer mapping.

Table A.578: 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*			
Attribute	*Type*	*Mul.*	*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.
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.579: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.580: 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: ▽

▽

△

Class	**SynchronizationTimingConstraint**			
	△ 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*			
Attribute	***Type***	***Mul.***	***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	1	attr	The specific type of this synchronization constraint.
tolerance	MultidimensionalTime	1	aggr	The maximum time interval, within which the synchronized events must occur.

Table A.581: 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*			
Attribute	***Type***	***Mul.***	***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.582: 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*			
Attribute	***Type***	***Mul.***	***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.

▽

△

Class	**System**			
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:** atpVariation **Tags:** vh.latestBindingTime=postBuild
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=shortName, 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=shortName, variationPoint.shortLabel vh.latestBindingTime=postBuild
pncVector Length	PositiveInteger	0..1	attr	Length of the partial networking request release information vector (in bytes).
pncVectorOffset	PositiveInteger	0..1	attr	Absolute offset (with respect to the NM-PDU) of the partial networking request release information vector that is defined in bytes as an index starting with 0.
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=shortName, variationPoint.shortLabel 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=shortName, variationPoint.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=-10
systemVersion	RevisionLabelString	1	attr	Version number of the System Description.

Table A.583: System

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*			
Attribute	***Type***	***Mul.***	***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.

Table A.584: SystemSignal

Class	**SystemSignalGroup**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	A signal group refers to a set of signals that must 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
systemSignal	SystemSignal	*	ref	Reference to a set of SystemSignals that must always be kept together.
transforming SystemSignal	SystemSignal	0..1	ref	Optional reference to the SystemSignal which shall contain the transformed (linear) data.

Table A.585: SystemSignalGroup

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.586: TDEventComplex

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*			
Attribute	***Type***	***Mul.***	***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	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 VariableDataPrototpyes in its expression. This association aggregates instance references to Variable DataPrototypes which can be referenced in the expression.

Table A.587: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
operation	ClientServerOperation	1	ref	The referenced operation.
tdEvent OperationType	TDEventOperationType Enum	1	attr	The specific type of this timing event.

Table A.588: TDEventOperation

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dataElement	VariableDataPrototype	1	ref	The referenced VariableDataPrototype
tdEventVariable DataPrototype Type	TDEventVariableData PrototypeTypeEnum	1	attr	The specific type of this timing event.

Table A.589: 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 Function Bus (VFB) level.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TimingDescription*, *TimingDescriptionEvent*			
Subclasses	*TDEventVfbPort*, TDEventVfbReference			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
component	SwComponent Prototype	0..1	iref	The context for the scope of this timing event.

Table A.590: 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 Function Bus (VFB) level.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TDEventVfb*, *TimingDescription*, *TimingDescriptionEvent*			
Subclasses	TDEventModeDeclaration, TDEventOperation, TDEventTrigger, TDEventVariableDataPrototype			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
isExternal	Boolean	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 Function 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.591: TDEventVfbPort

Class	**TagWithOptionalValue**			
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::TagWithOptionalValue			
Note	A tagged value is a combination of a tag (key) and a value that gives supplementary information that is attached to a model element. Please note that keys without a value are allowed. **Tags:** atp.ManifestKind=ServiceInstanceManifest			
Base	*ARObject*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
key	String	1	attr	Defines a key.
value	String	0..1	attr	Defines the corresponding value.

Table A.592: TagWithOptionalValue

Class	**TcpTp**
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology
Note	Content Model for TCP configuration.
Base	*ARObject*, *TcpUdpConfig*, *TransportProtocolConfiguration*

▽

△

Class	**TcpTp**			
Attribute	***Type***	***Mul.***	***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.
keepAliveTime	TimeValue	0..1	attr	Specifies the time in seconds between the last data packet sent and the first keepalive probe.
keepAlives	Boolean	0..1	attr	Indicates if Keep-Alive messages are send.
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 byte.
tcpTpPort	TpPort	1	aggr	TCP Port configuration.

Table A.593: 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*			
Attribute	***Type***	***Mul.***	***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	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	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.594: TextTableMapping

Class	**TextTableValuePair**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	Defines a pair of text values which are translated into each other.			
Base	*ARObject*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

▽

△

Class	**TextTableValuePair**			
firstValue	Numerical	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	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.595: 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*			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
value	VerbatimString	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.596: TextValueSpecification

Class	**TimeSyncClientConfiguration**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Defines the configuration of the time synchronisation client.			
Base	*ARObject*			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
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.597: TimeSyncClientConfiguration

Class	**TimeSyncServerConfiguration**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Defines the configuration of the time synchronisation server.			
Base	*ARObject*, *Referrable*			
Attribute	*Type*	*Mul.*	*Kind*	*Note*

▽

△

Class	**TimeSyncServerConfiguration**			
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.598: 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, *TDEventSwc*, *TDEventVfb*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
occurrence Expression	TDEventOccurrence Expression	0..1	aggr	The occurrence expression for this event.

Table A.599: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
response	TimingDescriptionEvent	1	ref	The response event representing the point in time where the event chain is terminated. **Tags:** xml.sequenceOffset=20
segment	TimingDescriptionEvent Chain	1..*	ref	A composed event chain consists of an arbitrary number of sub-chains. **Tags:** xml.sequenceOffset=30
stimulus	TimingDescriptionEvent	1	ref	The stimulus event representing the point in time where the event chain is activated. **Tags:** xml.sequenceOffset=10

Table A.600: TimingDescriptionEventChain

Class	**TimingEvent**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents			
Note	TimingEvent references the RunnableEntity that need to be started in response to the TimingEvent			
Base	*ARObject*, *AbstractEvent*, *AtpClassifier*, *AtpFeature*, *AtpStructureElement*, *Identifiable*, *MultilanguageReferrable*, *RTEEvent*, *Referrable*			
Attribute	***Type***	***Mul.***	***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	1	attr	Period of timing event in seconds. The value of this attribute shall be greater than zero.

Table A.601: TimingEvent

Class	***TimingExtension*** (abstract)			
Package	M2::AUTOSARTemplates::CommonStructure::Timing			
Note	The abstract parent class of the different template specific timing extensions. Depending on the specific timing extension (VfbTiming, SwcTiming, SystemTiming, BswModuleTiming, EcuTiming) the timing descriptions and timing constraints, that can be used to specify the timing behavior, are restricted.			
Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*			
Subclasses	BswCompositionTiming, BswModuleTiming, EcuTiming, SwcTiming, SystemTiming, VfbTiming			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
timingCondition	TimingCondition	*	aggr	The timing condition specifies a specific condition. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, 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=shortName, 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=shortName, variationPoint.shortLabel vh.latestBindingTime=postBuild

▽

△

Class	***TimingExtension*** (abstract)			
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=shortName, variationPoint.shortLabel vh.latestBindingTime=postBuild
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=shortName, variationPoint.shortLabel

Table A.602: 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. **Tags:** atp.ManifestKind=ServiceInstanceManifest atp.Status=draft			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
authentication	CryptoServicePrimitive	0..1	ref	This reference identifies the crypto service primitive for the generation and verification of MACs. **Tags:** atp.Status=draft
certificate	CryptoService Certificate	0..1	ref	This reference identifies the applicable certificate. **Tags:** atp.Status=draft
encryption	CryptoServicePrimitive	0..1	ref	This reference identifies the crypto service primitive for the execution of encryption. **Tags:** atp.Status=draft
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. **Tags:** atp.Status=draft
preSharedKey	CryptoServiceKey	0..1	ref	This reference identifies the applicable cryptograhic key if the handshake is based on the existence of a pre-shared key (PSK) **Tags:** atp.Status=draft
priority	PositiveInteger	0..1	attr	This attribute identifies the priority of the cipher suite. Range: 1..65535. Lower values represent higher priorities.
version	TlsVersionEnum	1	attr	This attribute supports the definition of the applicable version of TLS.

Table A.603: 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). **Tags:** atp.Status=draft atp.recommendedPackage=CryptoServiceMappings			
Base	*ARObject*, *CryptoServiceMapping*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
keyExchange	CryptoServicePrimitive	*	ref	This reference identifies the shared(i.e. applicable for each of the aggreated cipher suites) crypto service primitive for the execution of key exchange during the handshake phase. **Tags:** atp.Status=draft
tlsCipherSuite	TlsCryptoCipherSuite	*	aggr	This aggregation represents the collection of supported cipher suites. **Tags:** atp.Status=draft

Table A.604: TlsCryptoServiceMapping

Class	**TlvDataIdDefinition**			
Package	M2::AUTOSARTemplates::AdaptivePlatform::ApplicationDesign::SerializationProperties			
Note	This meta-class represents the ability to define the tlvDataId. **Tags:** atp.Status=draft			
Base	*ARObject*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
id	PositiveInteger	1	attr	This attribute represents the definition of the value of the TlvDataId
tlvArgument	ArgumentDataPrototype	0..1	ref	This reference assigns a tlvDataId to a given argument of a ClientServerOperation. **Tags:** atp.Status=draft
tlv Implementation DataType Element	ImplementationData TypeElement	0..1	ref	This reference associates the definition of a TLV data id with a given ImplementationDataTypeElement. **Tags:** atp.Status=draft
tlvRecord Element	ApplicationRecord Element	0..1	ref	This reference associates the definition of a TLV data id with a given ApplicationRecordElement. **Tags:** atp.Status=draft

Table A.605: TlvDataIdDefinition

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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

▽

△

Class	**Topic1**			
helpEntry	String	0..1	attr	This specifies an entry point in an online help system to be linked with the parent class. The syntax must be defined by the applied help system respectively help system generator. **Tags:** xml.attribute=true
topicContent	TopicContentOrMsrQuery	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.606: 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***	***Mul.***	***Kind***	***Note***
ident	TpConnectionIdent	0..1	aggr	This adds the ability to become referrable to Tp Connection.

Table A.607: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.608: TpConnectionIdent

Class	**TpPort**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Dynamic or direct assignment of a PortNumber.			
Base	*ARObject*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dynamicallyAssigned	Boolean	0..1	attr	Indicates whether the source port is dynamically assigned.
portNumber	PositiveInteger	0..1	attr	Port Number.

Table A.609: 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*, TraceableText			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
trace	Traceable	*	ref	This assocation 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.610: 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 appliles: • **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*			
Attribute	***Type***	***Mul.***	***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.611: TraceableText

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			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.612: 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, UserDefinedTransformationISignalProps			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
csErrorReaction	CSTransformerErrorReactionEnum	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.
dataPrototypeTransformationProps	DataPrototypeTransformationProps	*	aggr	Fine granular modeling of TransfromationProps on the level of DataPrototypes.
transformer	TransformationTechnology	1	ref	Reference to the TransformationTechnology description that contains transformer specific and ISignal independent configuration properties.

Table A.613: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
bufferProperties	BufferProperties	1	aggr	Aggregation of the mandatory BufferProperties.
hasInternalState	Boolean	0..1	attr	This attribute defines whether the Transformer has an internal state or not.
needsOriginalData	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.
transformationDescription	TransformationDescription	0..1	aggr	A transformer can be configured with transformer specific parameters which are represented by the Transformer Description. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
transformerClass	TransformerClassEnum	1	attr	Specifies to which transformer class this transformer belongs.
version	String	1	attr	Version of the implemented protocol.

Table A.614: TransformationTechnology

Class	**TransformerHardErrorEvent**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::RTEEvents			
Note	The 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*, *MultilanguageReferrable*, *RTEEvent*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
operation	ClientServerOperation	0..1	iref	This represents the ClientServerOperation to which the TransformerHardErrorEvent refers to.
trigger	Trigger	0..1	iref	Trigger for which the transformer can trigger this TransformerHardErrorEvent

Table A.615: TransformerHardErrorEvent

Class	**TransientFault**			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	The reported failure is classified as runtime error.			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*, *TracedFailure*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
possibleError Reaction	PossibleErrorReaction	*	aggr	Describes a possible error reactions for the transient fault handler.

Table A.616: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
timeout	TimeValue	1	attr	Number of seconds before an error is reported or in case of allowed redundancy, the value is sent again.

Table A.617: TransmissionAcknowledgementRequest

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*			
Attribute	***Type***	***Mul.***	***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.618: 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*			
Attribute	***Type***	***Mul.***	***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.619: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
trigger	Trigger	1..*	aggr	The Trigger of this trigger interface.

Table A.620: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
triggerMapping	TriggerMapping	1..*	aggr	Mapping of two Trigger in two different TriggerInterface

Table A.621: TriggerInterfaceMapping

Class	**TriggerPortAnnotation**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::ApplicationAttributes			
Note	Annotation to a port used for calibration regarding a certain Trigger.			
Base	*ARObject*, *GeneralAnnotation*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
trigger	Trigger	1	ref	The instance of annotated trigger.

Table A.622: 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*			
Attribute	***Type***	***Mul.***	***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

Table A.623: TriggerToSignalMapping

Class	**UdpNmCluster**			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	Udp specific NmCluster attributes **Tags:** atp.ManifestKind=MachineManifest			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *NmCluster*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
nmCbvPosition	Integer	0..1	attr	Defines the position of the control bit vector within the Nm Pdu (Byte positon).
nmChannel Active	Boolean	1	attr	This switch determines if the respective UdpNm channel is active or not. Indicates whether a particular UdpNm channel shall be initialized (TRUE) or not (FALSE). If this parameter is set to FALSE the respective NM instance shall not be used during runtime.
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	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	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 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.
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.

▽

△

Class	**UdpNmCluster**			
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.
vlan	EthernetPhysical Channel	0..1	ref	Reference to the vlan (represented by the Ethernet PhysicalChannel) this UdpNmCluster shall apply to.

Table A.624: UdpNmCluster

Class	**UdpTp**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Content Model for UDP configuration.			
Base	*ARObject*, *TcpUdpConfig*, *TransportProtocolConfiguration*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
udpTpPort	TpPort	1	aggr	Udp Port configuration.

Table A.625: UdpTp

Class	**UnassignFrameId**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	Schedule entry for an Unassign Frame Id master request where the protected identifier is assigned the value 0x40. This will disable reception/transmission of a previously dynamically assigned frame identifier.			
Base	*ARObject*, *LinConfigurationEntry*, *ScheduleTableEntry*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
messageId	PositiveInteger	0..1	attr	MessageId of the referenced frame.
unassigned FrameTriggering	LinFrameTriggering	1	ref	The frame whose identifier is reset by this assignment.

Table A.626: UnassignFrameId

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 (offsetSiToUnit) are applied as follows:
	`x [{unit}] := y * [{siUnit}] * factorSiToUnit [[unit]/{siUnit}] + of`
	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 [[u`
	Tags: atp.recommendedPackage=Units

Base	*ARElement*, *ARObject*, *CollectableElement*, *Identifiable*, *MultilanguageReferrable*, *Packageable Element*, *Referrable*

▽

△

Class	**Unit**			
Attribute	***Type***	***Mul.***	***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.627: 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=[+\-]?[1-9][0-9]*|0[xX][0-9a-fA-F]+|0[bB][0-1]+|0[0-7]* xml.xsd.type=string

Table A.628: 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*			
Attribute	***Type***	***Mul.***	***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.629: 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*			
Attribute	***Type***	***Mul.***	***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.630: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.631: 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			
Attribute	***Type***	***Mul.***	***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.632: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
accessed Variable	AutosarVariableRef	1	aggr	This denotes the accessed variable.

▽

△

Class	**VariableAccess**			
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.633: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dataMapping	DataPrototypeMapping	1..*	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.634: VariableAndParameterInterfaceMapping

Class	**VariableDataPrototype**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::DataPrototypes			
Note	A VariableDataPrototype is used to contain values in an ECU application. This means that most likely a VariableDataPrototype allocates "static" memory on the ECU. In some cases optimization strategies might lead to a situation where the memory allocation can be avoided. In particular, the value of a VariableDataPrototype is likely to change as the ECU on which it is used executes.			
Base	*ARObject*, *AtpFeature*, *AtpPrototype*, *AutosarDataPrototype*, *DataPrototype*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
initValue	ValueSpecification	0..1	aggr	Specifies initial value(s) of the VariableDataPrototype

Table A.635: VariableDataPrototype

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***	***Mul.***	***Kind***	***Note***
desc	MultiLanguageOverview Paragraph	0..1	aggr	This allows to describe shortly the purpose of the variation point. **Tags:** xml.sequenceOffset=20
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

▽

△

Class	**VariationPoint**			
formalBlueprint Condition	BlueprintFormula	0..1	aggr	This denotes a formal blueprintCondition. This shall be not in contradiction with blueprintCondition or formal BlueprintGenerator. It is recommended only to use one of the two. **Tags:** atp.Status=obsolete xml.sequenceOffset=29
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
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. **Tags:** xml.sequenceOffset=10
swSyscond	ConditionByFormula	0..1	aggr	This condition acts as Binding Function for the Variation Point. Note that the mulitplicity is 0..1 in order to support pure postBuild variants. **Tags:** xml.sequenceOffset=30

Table A.636: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
conditionAccess	ConditionByFormula	0..1	aggr	This condition acts as Binding Function for the Variation Point.
implementation DataType	ImplementationData Type	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.

▽

△

Class	**VariationPointProxy**			
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.637: VariationPointProxy

Class	**VfbTiming**			
Package	M2::AUTOSARTemplates::CommonStructure::Timing			
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*, *MultilanguageReferrable*, *PackageableElement*, *Referrable*, *TimingExtension*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
component	SwComponentType	1	ref	This defines the scope of a VfbTiming. All corresponding timing descriptions and constraints must be defined within this scope.

Table A.638: VfbTiming

Class	**VlanConfig**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	VLAN Configuration attributes			
Base	*ARObject*, *Identifiable*, *MultilanguageReferrable*, *Referrable*			
Attribute	***Type***	***Mul.***	***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.639: 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*			
Attribute	***Type***	***Mul.***	***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.

▽

△

Class	**VlanMembership**			
dhcpAddress Assignment	DhcpServer Configuration	0..1	aggr	Specifies the IP Address which will be assigned to a DHCP Client at this SwitchPort. If no dhcpAddressAssignment 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.640: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
timeout	TimeValue	1	attr	Time in seconds before the WaitPoint times out and the blocking wait call returns with an error indicating the timeout.
trigger	RTEEvent	1	ref	This is the RTEEvent this WaitPoint is waiting for.

Table A.641: WaitPoint

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*			
Attribute	***Type***	***Mul.***	***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
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

▽

△

Class	**Xdoc**			
url	Url	0..1	aggr	This specifies the URL of the external document. **Tags:** xml.sequenceOffset=70

Table A.642: Xdoc

Class	**Xfile**			
Package	M2::MSR::Documentation::TextModel::InlineTextElements			
Note	This represents to reference an external file within a documentation.			
Base	*ARObject*, *Referrable*, *SingleLanguageReferrable*			
Attribute	***Type***	***Mul.***	***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.643: 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*			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.644: XrefTarget