Document Title	System Template
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	063
Document Classification	Standard

Document Version	4.5.0
Document Status	Final
Part of Release	4.1
Revision	3

Document Change History			
Date	**Version**	**Changed by**	**Description**
31.03.2014	4.5.0	AUTOSAR Release Management	• Various fixes and clarifications
24.10.2013	4.4.0	AUTOSAR Release Management	• Set CanNmCluster.nmChannelActive, FlexrayArTpChannel.timeFrIf and FlexrayArTpChannel.maxFrIf to deprecated • Added SoAd Pdu Collection attributes to SocketConnection • Added SoAdRouting-Group.eventGroupControlType • Introduced SocketAd-dress.multicastConnector • Clarified usage of ISignal.dataTypePolicy • Described the handling of ComSpecs during flattening • Introduced new Pdu types: GeneralPurposePdu and GeneralPurposeIPdu

			• Made RootSwCompositionPrototype.calibrationParameterValueSet "atpSplitable" • Made RootSwCompositionPrototype.flatMap "atpSplitable" • Added new Ethernet addressing attributes to SocketConnection to help to derive the Ecu Configurations for the Server and the Clients
14.03.2013	4.3.0	AUTOSAR Administration	• Added support for remote activation of RunnableEntitys • Added support VLANs and Service Discovery • Reworked the SoAd configuration • Introduced SenderReceiverCompositeElementToSignalMapping and ClientServerToSignalMapping • Added support for CAN FD • Reworked the J1939 TP configuration • Clarification of the usage of swDataDefProps on ISignals and SystemSignals • Added support for Complex Drivers in the Topology • Updated IPduM to allow only static part reception • Added LinSlaveConfig class to the LinMaster • Clarified meaning of PduToFrameMapping.startPosition

08.11.2011	4.2.0	AUTOSAR Administration	• Added support for Partial Networking • Added support for Complex Drivers • Added support for new COM transfer properties • Added support for transmission mode switch via Com_SwitchIpduTxMode COM API • Added support for treating byte arrays with primitive type mapping • Added support for partial routing in signal gateways • Added support for FlexRay AUTOSAR TP • Added rules for creation of Pdu Triggerings and Pdu Ports • Explained the general approach of bit counting

22.10.2010	4.1.0	AUTOSAR Administration	• updated System class category names • Changed specification of PduLength parameter from bits to bytes • Made Flexray channel specific attributes optional • Clarified the usage of EcuPorts in System Extract/Ecu Extract • Allowed to define sending and receiving connections to EcuPorts for NmPdus, XcpPdus • Aligned FrTP model to AUTOSAR FrTp SWS • Replaced ComProcessingPeriod by three timebase parameters • Reworked E2E protection of selected I-PDUs • Corrected AssignFrameIdRange configuration in LIN model • Clarified the routing of ISignalGroups in the Signal Gateway • Extended the enumeration "TransferPropertyEnum" with the element "triggeredOnChange" • Added a subchapter to the appendix about special use cases that are supported by the System Template • Reworked SenderReceiverToSignalGroupMapping and ClientServerToSignalGroupMapping • Changed multiplicity between System and SystemMapping from 1 to 0..1.

04.12.2009	4.0.0	AUTOSAR Administration	• Implemented support for LIN 2.1 • Implemented support for Network Management (FlexRayNm, CanNm, LinNm, UdpNm) • adapted IPdu Multiplexer model to ASAM Fibex 3.1 • Reworked "ECU Extract" chapter • Introduced "System Extract" • Introduced EndToEndProtection for ISignalIPdus • Reworked "Transport Layer" chapter • Implemented Variant Handling concept • Implemented Documentation support concept • Implemented support for J1939 communication • Implemented support for TTCan • Implemented support for for TCP/IP and DoIP. • Introduced Pdu Counter and Pdu Replication • Implemented VMM/AMM concept • Introduced low-level routing of NPdu's • Implemented support for dynamic signals • Introduced PdurIPduGroups
26.01.2009	3.1.0	AUTOSAR Administration	• Clarified semantics of Data Mappings • Added inheritance from Identifiable to PduToFrameMapping • Added "FlexRayChannelName" attribute to FlexRayPhysicalChannel element.

03.06.2008	3.0.3	AUTOSAR Administration	• Added the boolean attribute "payloadPreambleIndicator" to the "FlexrayFrameTriggering". • Added extension that allows the assignment of IPduGroups to ECUs. • Added missing reference from "ClientServerComposite-TypeMapping" to "ArgumentPrototype" • Alignment with AUTOSAR IPduM SWS
15.02.2008	3.0.2	AUTOSAR Administration	• Legal disclaimer revised
31.01.2008	3.0.1	AUTOSAR Administration	• Moved "canAddressingMode" attribute from "CanCluster" to the "CanFrameTriggering" element • Clarified the descriptions of several elements and attributes.
04.12.2007	3.0.0	AUTOSAR Administration	• Communication part reworked from scratch • Alignment with ECU Configuration • Added support for Transport Protocols • Major changes in Topology chapter after harmonisation with Fibex (removed complex Topologies) • Document meta information extended • Small layout adaptations made

31.01.2007	2.0.0	AUTOSAR Administration	• Support for Signal Groups added. • Rework of the Topology Description • Introduction of PDUs. Description of the PDU Multiplexer, PDU Gateway. • FlexRay: multiple transmission of a frame within one communication cycle is supported now. • Removed the concept of Variant Descriptions (Properties) and CompToECUMappingConstraints relying on the property concept. • Split SwCompToEcuMapping in two classes in order to allow separation of SWC-to-ECU mapping and Implementation-to-SWC mapping. • Removed preliminary chapter on MOST as it is not part of the standard. • For all Instance References in the System Template added diagrams to the meta-model containing detailed representations of these references.
31.01.2007	2.0.0	AUTOSAR Administration	• Legal disclaimer revised • Release Notes added • "Advice for users" revised • "Revision Information" added
09.05.05	1.0.0	AUTOSAR Administration	Initial Release

Disclaimer

Advice for users

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

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

Table of Contents

References

[1] Requirements on System Template
AUTOSAR_RS_SystemTemplate

[2] Generic Structure Template
AUTOSAR_TPS_GenericStructureTemplate

[3] Model Persistence Rules for XML
AUTOSAR_TR_XMLPersistenceRules

[4] Methodology
AUTOSAR_TR_Methodology

[5] Software Component Template
AUTOSAR_TPS_SoftwareComponentTemplate

[6] Specification of ECU Resource Template
AUTOSAR_TPS_ECUResourceTemplate

[7] Standardization Template
AUTOSAR_TPS_StandardizationTemplate

[8] Specification of Timing Extensions
AUTOSAR_TPS_TimingExtensions

[9] ASAM Fibex - Field Bus Exchange Format, Version 3.1
http://www.asam.net

[10] LIN Specification Package, Version 2.1
http://www.lin-subbus.org

[11] CAN specifications
http://www.can-cia.org

[12] MOST Specification, Version 2.5
http://www.mostnet.de

[13] FlexRay Protocol Specification
http://www.flexray.com

[14] Layered Software Architecture
AUTOSAR_EXP_LayeredSoftwareArchitecture

[15] Specification of LIN Interface
AUTOSAR_SWS_LINInterface

[16] Specification of RTE Software
AUTOSAR_SWS_RTE

[17] Basic Software Module Description Template
AUTOSAR_TPS_BSWModuleDescriptionTemplate

[18] Specification of SW-C End-to-End Communication Protection Library
AUTOSAR_SWS_E2ELibrary

[19] Specification of Communication
AUTOSAR_SWS_COM

[20] Specification of I-PDU Multiplexer
AUTOSAR_SWS_IPDUMultiplexer

[21] ASAM MCD 2MC ASAP2 Interface Specification
http://www.asam.net
ASAP2-V1.51.pdf

[22] Software Process Engineering Meta-Model Specification
http://www.omg.org/spec/SPEM/2.0/

1 Introduction

1.1 Abbreviations

Abbreviation	*Meaning*
CAN	Controller Area Network
CAS	Collision Avoidance Symbol
CBV	Control Bit Vector
CC	Communication Controller
DoIp	Diagnostics over IP
DTD	Document Type Definition
ECU	Electrical Control Unit
FIBEX	Field Bus Exchange Format
I^2C	Inter-Integrated Circuit
ID	Identifier
IPDU	Interaction Layer Protocol Data Unit
ISG	Inter-slot Gap
LIN	Local Interconnect Network
LPDU	Data Link Layer Protocol Data Unit
MOST	Media Oriented Systems Transport
NAD	Node Address for Diagnostic
NID	NOde Identification
NIT	Network Idle Time
NM	Network Management
NPDU	Network Layer Protocol Data Unit
OBD	Onboard Diagnostic
PDU	Protocol Data Unit
POC	Protocol Operation Control
RTE	Runtime Environment
SDU	Service Data Unit
SID	Service Identifier
SPI	Serial Peripheral Interface
SWC	Software Component
SWC-T	Software Component Template
SYS-T	System Template
TP	Transport Protocol
TTCAN	Time Triggered Controller Area Network
UML	Unified Modeling Language
VFB	Virtual Functional Bus
XML	Extensible Markup Language
XSD	XML Schema Definition

1.2 Requirements Tracing

The following table references the requirements specified in [1] and links to the fulfillment of these.

Requirement	Description	Satisfied by
[RS_SYST_00001]	Mixed Systems (AUTOSAR/NON-AUTOSAR)	[TPS_SYST_01063] [TPS_SYST_05000]
[RS_SYST_00002]	Basic Software Resources and RTE Resources	[TPS_SYST_01126]
[RS_SYST_00003]	Iterative Development	[TPS_SYST_01000] [TPS_SYST_01002] [TPS_SYST_01003]
[RS_SYST_00006]	Compatibility between the AUTOSAR Templates	[TPS_SYST_01017] [TPS_SYST_01019]
[RS_SYST_00007]	Mapping of Software Components to ECUs	[TPS_SYST_01001] [TPS_SYST_01020] [TPS_SYST_01021] [TPS_SYST_01022]
[RS_SYST_00008]	SWC Cluster	[TPS_SYST_01024] [TPS_SYST_01025]
[RS_SYST_00009]	SWC Separation	[TPS_SYST_01026] [TPS_SYST_01045]
[RS_SYST_00010]	Exclusive Mapping of SWCs	[TPS_SYST_01029]
[RS_SYST_00011]	Dedicated Mapping of SWCs	[TPS_SYST_01027]
[RS_SYST_00013]	Topology	[TPS_SYST_01004] [TPS_SYST_01005] [TPS_SYST_01006] [TPS_SYST_01007] [TPS_SYST_01008] [TPS_SYST_01009] [TPS_SYST_01010] [TPS_SYST_01011] [TPS_SYST_01013] [TPS_SYST_01014] [TPS_SYST_01015]
[RS_SYST_00014]	Data Segmentation	[TPS_SYST_01099] [TPS_SYST_01100] [TPS_SYST_01101] [TPS_SYST_01102] [TPS_SYST_01103] [TPS_SYST_01104] [TPS_SYST_01105] [TPS_SYST_01106]
[RS_SYST_00016]	Dedicated physical connections	[TPS_SYST_01043]
[RS_SYST_00017]	Mapping of signals to the same physical line	[TPS_SYST_01041]
[RS_SYST_00018]	Mapping of signals to different physical lines	[TPS_SYST_01044]
[RS_SYST_00019]	Mapping of signals to a specific physical line	[TPS_SYST_01043]
[RS_SYST_00020]	Exclusion of signals from a specific physical line	[TPS_SYST_01042]
[RS_SYST_00021]	ECU Communication via CAN	[TPS_SYST_01130]

Requirement	Description	Satisfied by
[RS_SYST_00022]	ECU Communication via LIN	[TPS_SYST_01012] [TPS_SYST_01129]
[RS_SYST_00024]	ECU Communication via FlexRay	[TPS_SYST_01085] [TPS_SYST_01128]
[RS_SYST_00025]	Derivation of COM Stack Configuration Parameters from the System Template	[TPS_SYST_01030]
[RS_SYST_00027]	ECU Extract generation rules	[TPS_SYST_01000] [TPS_SYST_01002] [TPS_SYST_01003] [TPS_SYST_01016]
[RS_SYST_00028]	IPdu End-to-End Communication Protection support	[TPS_SYST_01070] [TPS_SYST_01071] [TPS_SYST_01072] [TPS_SYST_01073] [TPS_SYST_01074]
[RS_SYST_00029]	Dynamic length signals	[TPS_SYST_01049] [TPS_SYST_01065]
[RS_SYST_00030]	Dynamic length IPdus	[TPS_SYST_01049]
[RS_SYST_00031]	Distribution of Application and Vehicle Mode Requests	[TPS_SYST_01023]
[RS_SYST_00033]	Software-to-ECU mapping variants	[TPS_SYST_01001]
[RS_SYST_00037]	Timing properties	[TPS_SYST_01075] [TPS_SYST_01076] [TPS_SYST_01077]
[RS_SYST_00038]	Support of SAE J1939 Protocol Features	[TPS_SYST_01106] [TPS_SYST_01132]
[RS_SYST_00039]	ECU Communication via Ethernet	[TPS_SYST_01086] [TPS_SYST_01088] [TPS_SYST_01089] [TPS_SYST_01090] [TPS_SYST_01091] [TPS_SYST_01092] [TPS_SYST_01093] [TPS_SYST_01094] [TPS_SYST_01095] [TPS_SYST_01096] [TPS_SYST_01097] [TPS_SYST_01098] [TPS_SYST_01108] [TPS_SYST_01131]
[RS_SYST_00042]	Support for Partial Networking	[TPS_SYST_01133]
[RS_SYST_00043]	Communication via Complex Drivers	[TPS_SYST_01115]
[RS_SYST_00044]	Description of custom bus systems	[TPS_SYST_01127]
[RS_SYST_00045]	Co-existing System artifacts in the same model	[TPS_SYST_03000]
[RS_SYST_00047]	Network and physical representation on signal level	[TPS_SYST_01062] [TPS_SYST_01063]
[RS_SYST_00048]	CAN with Flexible Data-Rate	[TPS_SYST_01154]

1.3 Requirements not fulfilled by TPS requirements

This section contains a list of requirements that are not yet fulfilled by TPS requirements.

Requirement	Description	Satisfied by
[RS_SYST_00015] Bus bandwidth	The System Template shall support bandwidth calculation as a constraint for the definition of the Communication Matrix.	chapter Topology (3); Communication (chapter 6)
[RS_SYST_00023] ECU Communication via MOST	The System Template has to cover the system communication via MOST.	not covered
[RS_SYST_00025] Derivation of ECU Configuration Parameters from the System Template	The System Template shall enable the configuration of the Com Stack of the ECU. It handles those parameters that are necessary to describe the inter-ECU communication. Configuration parameters local to an ECU are not in the scope of the System Template.	Harmonization between Upstream Templates and ECU Configuration (chapter D)
[RS_SYST_00026] Fibex compatibility	Whenever there is a considerable overlap between the System Template and the ASAM FIBEX Standard, the System Template shall adopt the structures of the ASAM FIBEX Standard.	AUTOSAR System Template and ASAM FIBEX (chapter 1.8)
[RS_SYST_00032] Topology Variants	The System Template shall provide the means to describe topology variants with optional/alternative ECUs and communication clusters.	chapter Variant Handling 1.7.2 and chapter Topology 3.
[RS_SYST_00033] Software-to-ECU mapping variants	The System Template shall provide the means to describe alternative mappings of software components to ECUs.	chapter 1.7.2 Variant Handling and chapter 5.1 Software Component Mapping.
[RS_SYST_00034] Timing variants	The System Template shall provide the means to describe alternative timing properties (e.g. trigger type, period, priority) and timing constraints (e.g. latency, age).	chapter 1.7.2 Variant Handling and chapter 6 Communication.
[RS_SYST_00035] Data mapping variants	The System Template shall provide the means to describe data mapping Variants.	chapter 1.7.2 Variant Handling and chapter 5.2 Data Mapping.
[RS_SYST_00036] Communication variants	The System Template shall provide the means to describe communication variants, such as alternative signal-to-PDU mappings, alternative communication paths, and alternative signal and PDU properties (e.g. data type, data length).	chapter 1.7.2 Variant Handling and chapter 6 Communication.
[RS_SYST_00040] Timing constraints	The System Template shall provide the means to describe the timing constraints of a system's dynamics, which are determined by the consumption of computation, communication, and other hardware resources.	Timing Extensions (chapter 1.7.3)

Requirement	Description	Satisfied by
[RS_SYST_00041] Variants in ECU Extract	The ECU Extract shall support variability of elements taken over or derived during the transformation from the System Description.	Variant Handling in ECU Extract (chapter 10.6)

1.4 Methodology for Defining Formal Template

Figure 1.1 illustrates the overall methodology used to define formal templates. As is explained in the "Generic Structure Template" [2], it is important to separate a precise and concise model of the information that needs to be captured from the concrete XML-DTDs, XML-Schemas or other technology that is used to define the actual templates.

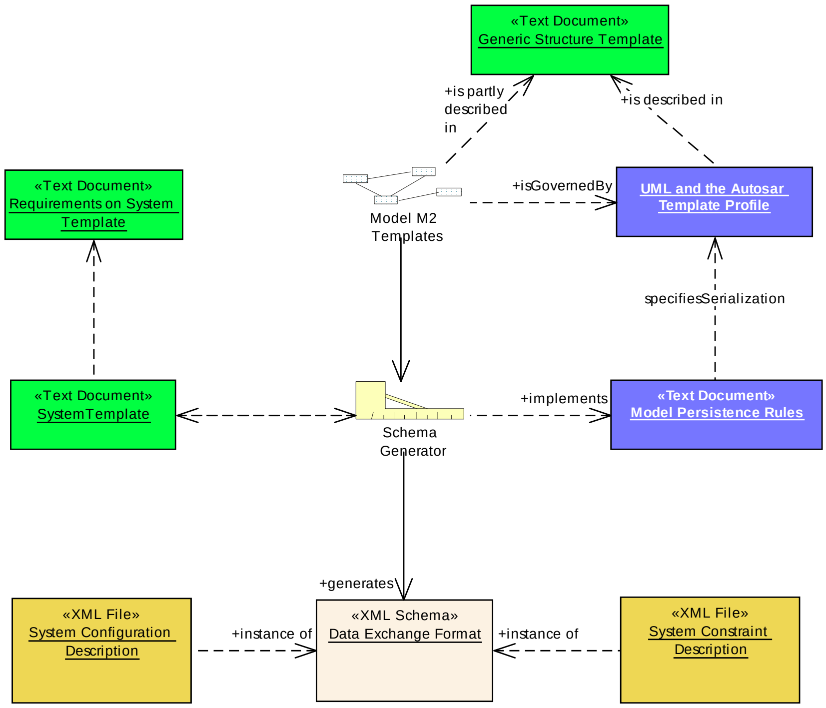

Figure 1.1: Methodology to define templates in AUTOSAR

The following documents describe the various aspects of the methodology:

1. The document called System Template (this document) describes the information that can be captured in the "system constraint" and "system configuration" description, independently from the mapping of this model on XML-technology. This document is based upon the AUTOSAR meta-model and contains an elaborate description of the semantics (the precise meaning) of all the information that can be captured within the relevant parts of this meta-model.

2. The UML and the AUTOSAR Template Profile [2] describes the basic concepts that should be used when creating content of the meta-model.

3. The document called "Model Persistence Rules for XML" [3] describes how XML is used and how the meta-model designed in the "System Template" should be translated by the "Schema Generator" (MMT) into XML-Schema (XSD) "Data Exchange Format". This "formalization strategy" is to be used for all data that is formally described in the meta-model. In particular this document is worth to read in order to understand the mapping of the meta-model and the XML based System template.

4. The "Generic Structure Template" [2] describes the top level structure which is common to all AUTOSAR templates and provides AUTOSAR standard mechanisms of modeling elements and patterns.
5. The concrete "Template", the "Data Exchange Format" is an XML schema which is generated out of the meta-model described in the "System Template" using the approach and the patterns defined in the "Model Persistence Rules for XML". This schema is typically used as input to tools. The M1-level system descriptions are XML files which can be validated against the schema. In that sense they are instances of the schema defining the XML representation of the template.

1.5 Scope

This document describes the system template and its use for the System Constraint Description and the System Configuration Description. In general a filled system template defines the relationship between the pure Software View on the System (represented by a top level SW Component Composition) and a Physical System Architecture with networked ECU instances. The system template is used in two stages of the "AUTOSAR Methodology" [4] (see Figure 1.2).

- As System Constraint Description it serves as input to the AUTOSAR system generator
- As System Configuration Description it defines the output of the AUTOSAR System Configuration Generator and serves as input to the AUTOSAR ECU Configuration Generator for the different ECUs defined in the description.
- As ECU Extract of the System Configuration Description it describes the ECU specific view on the System Description. It is individually generated for each of the System's ECU as the output of the AUTOSAR ECU Configuration Generator.

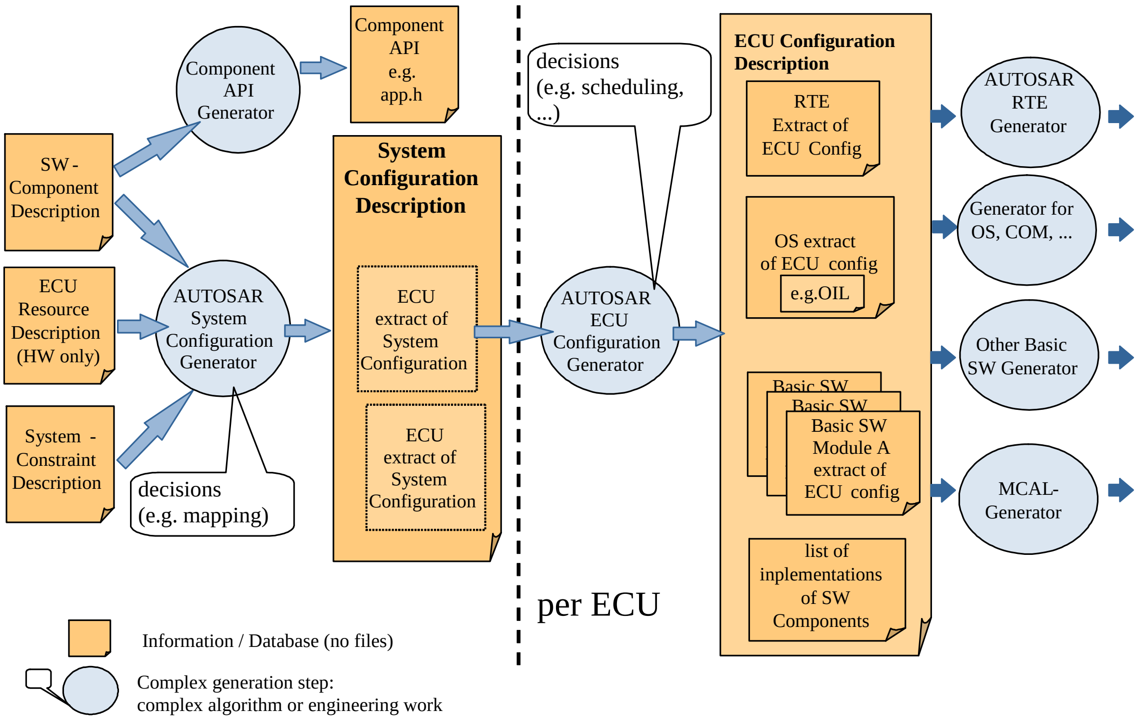

Figure 1.2: AUTOSAR Methodology

The System Template defines five major elements: Topology, Software, Communication, Mapping and Mapping Constraints, which will be defined in detail in the following chapters. Figure 1.3 gives an overview how these are used in the two different descriptions.

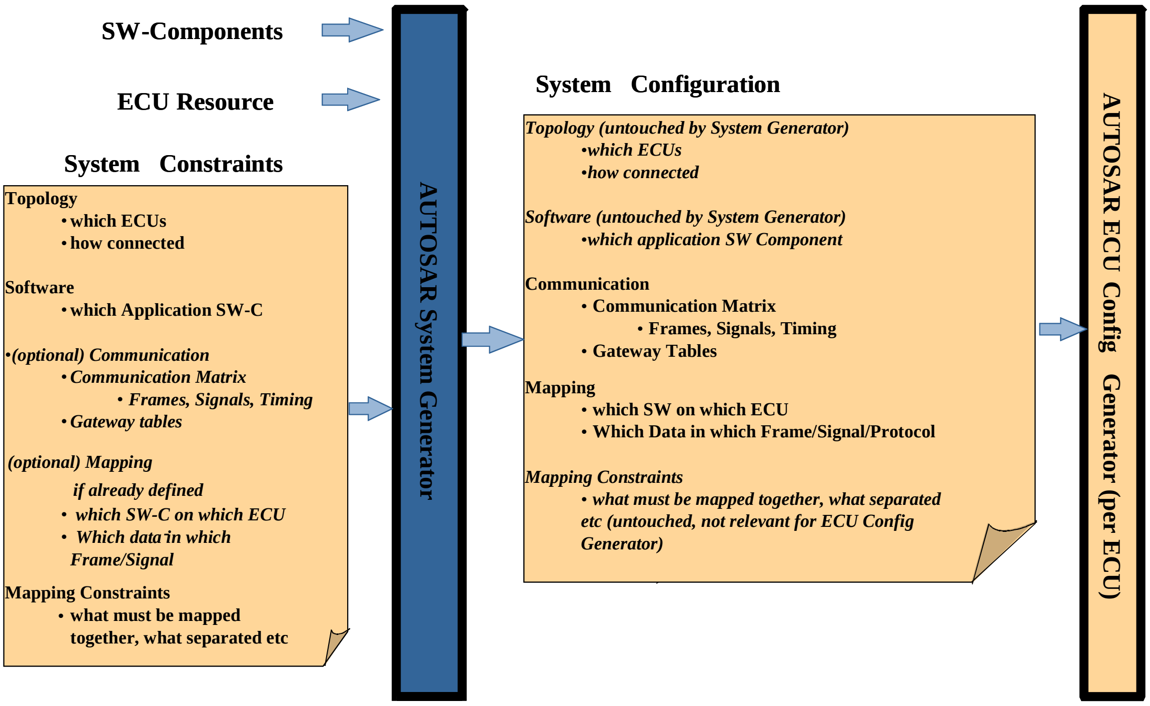

Figure 1.3: Scope of System Constraint Description and System Configuration Description

On Figure 1.3 some of the elements are marked `optional` for the System Constraint Description. If one starts with a new AUTOSAR project, these elements may not be present in the System Constraint Description. No (at least partial) functionality has been mapped yet, thus the communication matrix is not populated. But in most cases, many functional mappings are already predefined and contribute to the population of the communication matrix with their associated signals, thus being present in the System Constraint Description.

Reasons for such a predefinition are manifold. In some cases, hardware setup dictates where certain functionality resides, in some cases, a partial or complete communication matrix and/or completely configured ECUs (HW and SW) of another system (vehicle) has to be taken over. This approach is eased by the fact that System Configuration and System Constraint Description use the same format. That way it is possible to reuse parts of a System Configuration Description of the other system/vehicle in the actual System Constraint Description.

Furthermore, in the figure some of the elements are marked `untouched` for the System Configuration Description. This can have two reasons:

- The System Generator does not modify neither the Topology (networked ECUs) nor the Software, so these parts are just moved from System Constraint Description to System Configuration Description during the generation step.
- In a completed System Configuration Description, all SW components and all ECU-to-ECU communication have been mapped. Thus mapping constraints that limit the flexibility in the mapping phase of the system generator are obsolete

and will not be used in subsequent generator steps. They may however still be present for documentation and validation reasons.

Even if the communication matrix is determined as the result of the system configuration, the ECUs still have to be configured. This is done by the ECU configuration generator, which takes the System Configuration description as input and generates the ECU configuration description. The following guiding principles have been used to determine which information must be part of the System Configuration Description and which goes into the ECU Configuration Description:

- Information that is common for several ECUs and has to be agreed, must be part of the System Configuration Description and is thus covered by the System Template.
- Information, that only has ECU-local relevance is part of the ECU Configuration Description.

Thus the ECU Configuration Description will include the OS-schedule, the RTE-configuration and last but not least the configuration of the ECU basic software including the concrete communication drivers on that ECU.

1.6 UML Meta-Model

This chapter gives an overview of the AUTOSAR Unified Modeling Language (UML) meta-model. All AUTOSAR templates use a common meta-model. The templates describe software components, ECU resources, the Basic Software Modules, the ECU Configuration Parameters (ECU Configuration Description and ECU Configuration Parameter Definition) and the System.

The System Template defines all elements, their parameters and their relations, which are necessary for the System Constraint Description and the System Configuration Description.

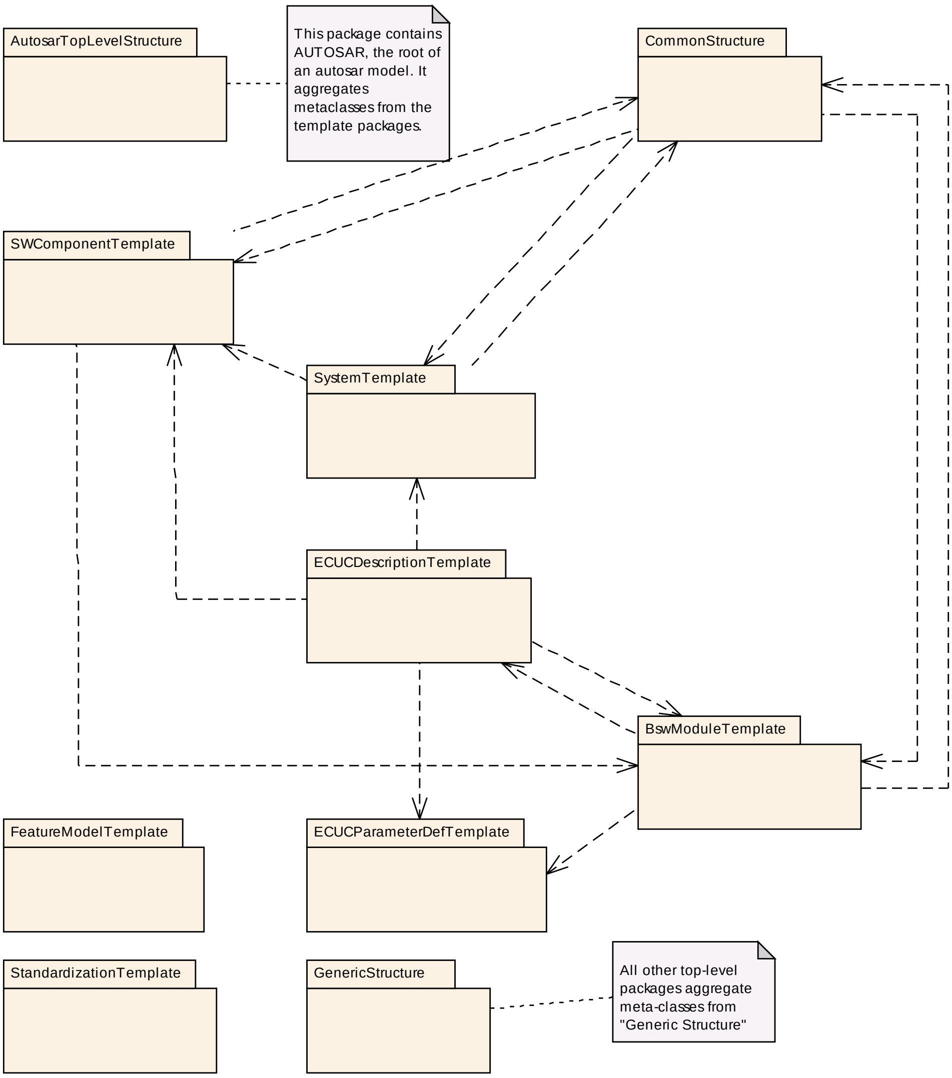

Figure 1.4: AUTOSAR Package Overview

Figure 1.4 shows the overall structure of the meta-model.

The dashed arrows in the diagram describe dependencies in terms of import-relationships between the packages within the meta-model. For example, the package `SystemTemplate` imports meta-classes defined in the packages `GenericStructure` [2], `SWComponentTemplate` [5] and `ECUResourceTemplate` [6].

For clarification, please note that the package `GenericStructure` contains some fundamental infrastructure meta-classes and common patterns that are described in [2]. As these are used by all other template specification the dependency associations are not depicted in the diagram for the sake of clarity.

Generic Structure provides details about

- Autosar Top level structure,
- Commonly used metaclasses and primitives
- Variant Handling

- Documentation

The ECU Resource Template deals with the description of the hardware resources of an ECU. The collection of all ECUs, which are integrated in the car, are described in the topology part of the System Configuration Description/System Constraint Description. Each of these ECUInstances uses the ECU Resource Template to describe the hardware resources. That's the reason, why the topology part has references to the ECU Resource Description.

The SW component description describes the SW components as well as their communication by data elements. The top-level software composition (RootSwCompositionPrototype) is part of the System Template (Software). This top-level software composition contains the functionality of the full system and describes the complete application software architecture of this system. The definition of the top level software composition uses the elements defined in the SW Component Template, like e.g. SwComponentType, PortInterface, AssemblySwConnector and DelegationSwConnector. That's why the System Description has references to the Software Component Description. The top level software composition is described in more detail in chapter 4.

Every template starts with an element AUTOSAR. While the models created in accordance to this guide are independent of the used formalization, it may still help the reader's understanding to note that AUTOSAR would also typically be the root element of a XML Schema generated from such a model. AUTOSAR can then contain one or more nested packages, simply allowing to further structure the contents of the M1 model[1].

1.7 Document Conventions

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

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

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

[1] A model and its meta-model are said to be on different meta levels (also referred to as abstraction levels). In AUTOSAR a five layer meta-model hierarchy is used, consisting of the five meta levels M0, M1, M2, M3 and M4 where entities in M0 are expressed in terms of M1 entities, M1 is expressed in terms of M2 entities and so on. The AUTOSAR meta-model hierarchy is described in more detail in the Autosar Template Modeling Guide [2].

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

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

Class	**AUTOSAR**			
Package	M2::AUTOSARTemplates::AutosarTopLevelStructure			
Note	Root element of an AUTOSAR description, also the root element in corresponding XML documents. **Tags:** xml.globalElement=true			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
adminData	AdminData	0..1	aggr	This represents the administrative data of an Autosar file. **Tags:** xml.sequenceOffset=10
arPackage	ARPackage	*	aggr	This is the top level package in an AUTOSAR model. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variation Point.shortLabel vh.latestBindingTime=blueprintDerivationTime xml.sequenceOffset=30
introduction	Documentation Block	0..1	aggr	This represents an introduction on the Autosar file. It is intended for example to rpresent disclaimers and legal notes. **Tags:** xml.sequenceOffset=20

Table 1.3: AUTOSAR

The first rows in the table have the following meaning:

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

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

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

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

The headers in the table have the following meaning:

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

Datatype: The datatype of an attribute of the class.

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

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

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

The verbal forms for the expression of obligation specified in [TPS_STDT_00053] shall be used to indicate requirements, see Standardization Template, chapter Support for Traceability ([7]).

The representation of requirements in AUTOSAR documents follows the table specified in [TPS_STDT_00078], see Standardization Template, chapter Support for Traceability ([7]).

1.7.1 Detailed Representation of InstanceRef Associations

As a special type of association "instanceRef" refers to an exact instance of the referenced class, requiring additional information of the target and the context. This is explained in detail in the AUTOSAR Generic Structure Template [2]. Each "instanceRef" association can both be represented by the short form and by an detailed representation. For readability the diagrams in the main body of the specification use the short form. The detailed descriptions can be found in the Appendix C.

1.7.2 Variant Handling

The System Template supports the creation of Variants in many of its model elements. In the Metamodel all locations that may exhibit variability are marked with the stereotype `atpVariation`. This allows the definition of possible variation points. Tagged Values are used to specify additional informations.

There are four types of locations in the metamodel which may exhibit variability:

- Aggregations
- Associations
- Attribute Values
- Classes providing property sets

The reasons for the attachment of the stereotype `atpVariation` to certain model elements and the consequences for other model elements are explained in class tables in the following chapters. More details about the AUTOSAR Variant Handling Concept can be found in the AUTOSAR Generic Structure Template [2].

1.7.3 Timing Extensions

With AUTOSAR Release 4.0 a new set of concepts for the description and analysis of end-to-end timing constraints is introduced by the Specification of Timing Extensions. A subset of these extensions aims for the system level and can be used to enhance the descriptions that are already available in the System Template.

A dedicated description of the timing extensions that can be used at system level is given in chapter 3 (System timing) in the Specification of Timing Extensions [8].

1.7.4 Documentation Support

With AUTOSAR Release 4.0 the AUTOSAR XML schema provides support for integrated and well structured documentation. More details about the AUTOSAR Documentation Support concept can be found in the AUTOSAR Generic Structure Template [2]. An optional documentation block can be applied to any identifiable element. Furthermore, as shown in figure 1.5, the System Template provides the possibility of adding additional documentation to several non-identifiable elements. The documentation of a `System` is composed of several chapters.

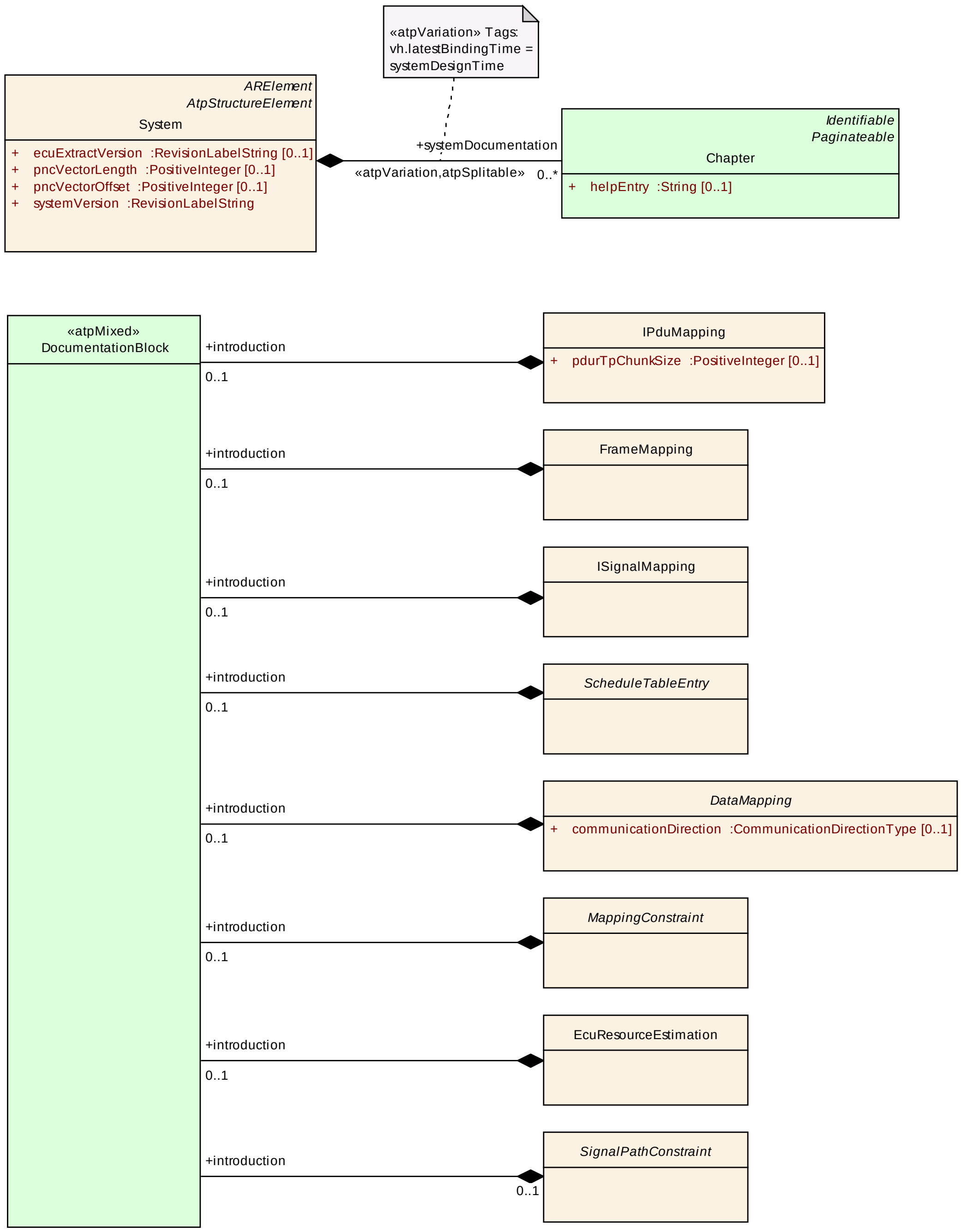

Figure 1.5: System Template Documentation Support

1.7.5 Stereotype atpSplitable in the System Template

The stereotype ≪atpSplitable≫ is used in the System Template to support step-wise processes, where the System Configuration Description is completed incrementally over a development process. Example:

1) Description of Communication only consists of interaction signals (ISignal). This is enough information to create an individual ECU's RTE, and even contains enough information to configure an ECU where the actual Frame/Pdu communication is being handled post-build.

2) In a second step, the communication matrix is being completed for a concrete vehicle. Pdus and Frames, along with their Triggerings are being added to the previous System Description. This model then contains the full information about an ECU's communication, especially containing the additional information to generate the post build information.

So, in this 2-step approach, an OEM could deliver the incomplete ECU extract from step (1) to the ECU integrator, who can then build a complete software image for the ECU. In the 2nd step, the ECU extract will be completed by the previously missing information, but as the first extract will still be valid due to the ≪atpSplitable≫ construct, the ECU including the flashed image from step (1) can be (re)used as it is, and just will be completed with the post build information, e.g. Frames and Pdus.

Further details about the ≪atpSplitable≫ stereotype can be found in the Generic Structure Template [2].

1.8 AUTOSAR System Template and ASAM FIBEX

FIBEX (Field Bus Exchange Format) [9] is an XML exchange format proposed for data exchange between tools that deal with bus communication Systems. The format supports the most common automotive data buses: LIN [10], CAN [11], MOST [12], FlexRay [13]. The covered areas of the exchange format are the functional network, system topology and the communication level. The functional network describes the software architecture of the system. In the system topology the logical layout of the system is described. This means it is documented which ECU is connected to which bus. The central purpose of a communication system is the exchange of frames with certain properties. The format is able to describe frames and their timing properties.

In future versions of the System Template a common subset between ASAM Fibex and Autosar will be harmonized. The current version of the System Template contains already the ASAM FIBEX description for communication and topology. Due to requirements of AUTOSAR some extensions were made to those descriptions. For instance the communication part is extended by a concept for PDUs (I-Pdus and N-Pdus). The harmonization between ASAM Fibex and AUTOSAR System Template is not finalized at this time.

In the UML Meta-Model the FIBEX contents are located in an own FIBEX UML Package. The top level `FibexElement` is referenced by the top level element `System` of the System Template. Similar to the usage of the `ARElement`, specializations of the `FibexElement` represent elementary building blocks within the FIBEX package. Each of this elements will be described in more detail in the following chapters.

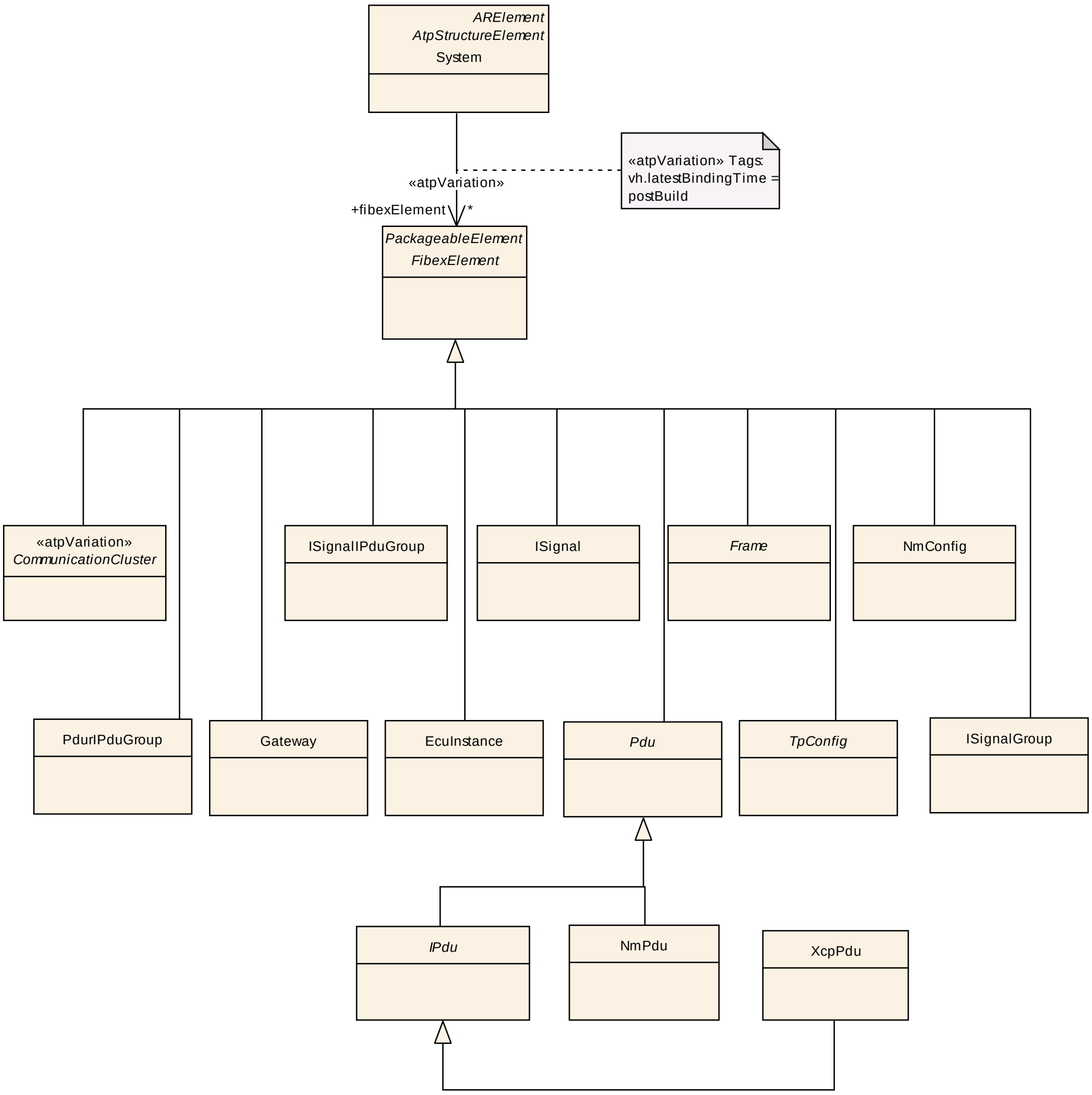

Figure 1.6: Fibex Elements

2 System

The top level element of the System Template is the class System, as shown in figure 2.1.

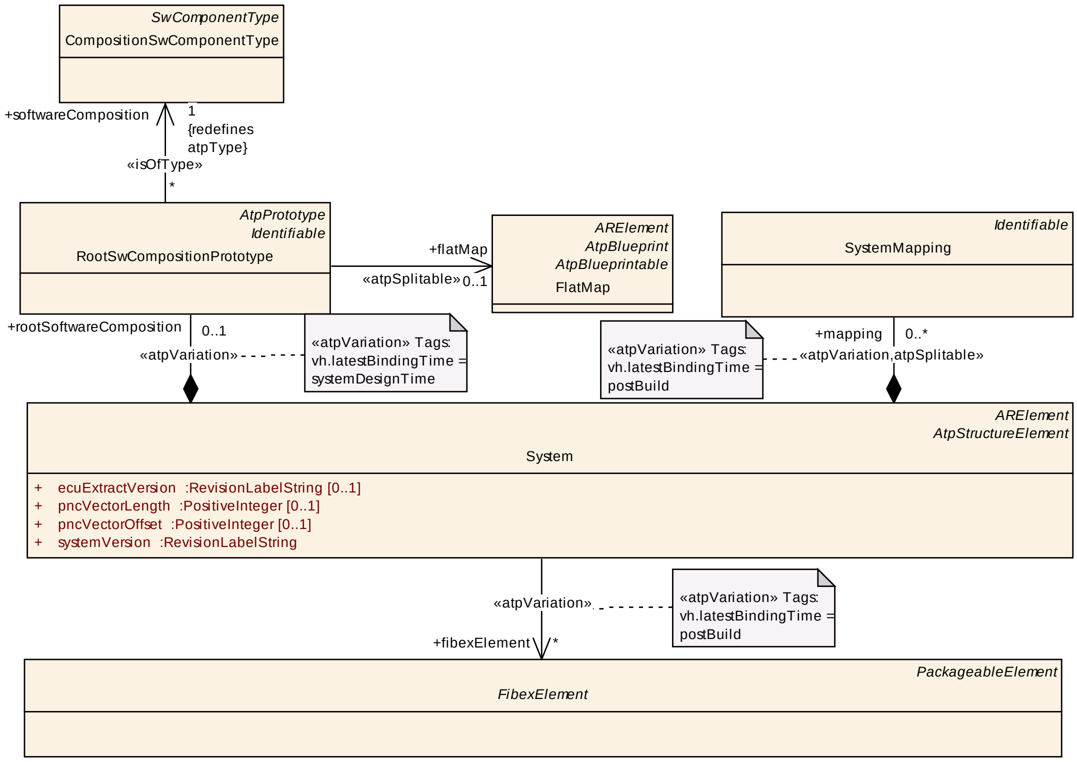

Figure 2.1: System Template Overview

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,Collectable Element,Identifiable,MultilanguageReferrable,PackageableElement,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
ecuExtract Version	RevisionLabelSt ring	0..1	attr	Version number of the Ecu Extract.

Attribute	Datatype	Mul.	Kind	Note
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 FibexElements can be optional. **Stereotypes:** atpVariation **Tags:** 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 SystemMapping 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
pncVectorLength	PositiveInteger	0..1	attr	Length of the partial networking request release information vector (in bytes).
pncVectorOffset	PositiveInteger	0..1	attr	Absolute offset (with respect to the NM-PDU) of the partial networking request release information vector that is defined in bytes as an index starting with 0.
rootSoftwareComposition	RootSwCompositionPrototype	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:** atpVariation **Tags:** vh.latestBindingTime=systemDesignTime

Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
systemDocumentation	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 2.1: System

`System` has relationships to all elements that define a system constraint description or system configuration description. It aggregates the `SystemMapping` and `RootSwCompositionPrototype` elements. `SystemMapping` deals with mapping of software components to ECUs as well as with the mapping of data elements that are to be exchanged between software components onto signals and frames. The `RootSwCompositionPrototype` element contains a reference to the top level software composition.

[constr_3028] **`FibexElements`** ⌈ Each `FibexElement` that is used in the System Description shall be referenced by the `System` element in the role `FibexElement`. ⌋

`FibexElement`s can be defined in a stand alone and reusable way (hence they can simply be created in any package like ARElements), but on the other hand it shall be clear that a certain `FibexElement` actually belongs to a certain System Description. Thus, all `FibexElement`s used within a System Description (i.e. contributing to the specification of the System communication and topology) shall be referenced from the `System` element. More details about the integration of FIBEX into the System Template will be given in chapter 1.8.

[TPS_SYST_01002] **`System Category`** ⌈ The `System` shall have a `category` element defined which indicates the role of this work product. ⌋*(RS_SYST_00003, RS_SYST_00027)*

[TPS_SYST_01003] **`Standardized System Category Definitions`** ⌈

category	*Meaning*
SYSTEM_CONSTRAINTS	The `System` class is used to describe System Constraints. In this usage, it forms the core element of a System Constraints Description, serving as an input to the AUTOSAR System Generator.
SYSTEM_DESCRIPTION	The `System` class is used to describe the System Configuration of a complete AUTOSAR System. In this usage, it forms the core element of a System Description, the output of the AUTOSAR System Generator.
SYSTEM_EXTRACT	The `System` class is used to describe a subsystem specific view on the complete System Description. The System Extract is not fully decomposed and still contains compositions. The SYSTEM_EXTRACT is the basis for designing subsystems.

category	*Meaning*
ECU_EXTRACT	The System class is used to describe the ECU specific view on the complete System Description. In this usage, it forms the core element of ECU Extract, the output of the AUTOSAR ECU Configuration Extractor. The ECU Extract is fully decomposed and contains only atomic software components. The ECU Extract is the basis for setting up the ECU Configuration.
ABSTRACT_SYSTEM_DESCRIPTION	This System is used to describe a functional (solution-independent/abstract) system design. It can be taken as basis for the development of the SYTEM_DESCRIPTION. No structural constrains are applied on the transformation of the ABSTRACT_SYSTEM_DESCRIPTION to the SYSTEM_DESCRIPTION.
ECU_SYSTEM_DESCRIPTION	This System is used to describe the closed view on one ECU (note that an AUTOSAR ECU is defined being one microprocessor running one AUTOSAR Stack). It can be derived from a SYSTEM_EXTRACT or it can be designed independently and mapped to a SYSTEM_EXTRACT. The ECU_SYSTEM_DESCRIPTION is not fully decomposed and still may contain compositions.

Table 2.2: System class categories

⌋*(RS_SYST_00003, RS_SYST_00027)*

Note: SYSTEM_EXTRACT does not prescribe the number of micro controllers / cores for one ECU from the OEM perspective.

- Supplier decides to design one AUTOSAR ECU with multicore support leads to one ECU_EXTRACT supporting one AUTOSAR stack
- Supplier decides to design two AUTOSAR ECUs in one box leads to two ECU_EXTRACTs supporting two AUTOSAR stacks

[constr_3027] Existence of `ecuExtractVersion` ⌈In case the category of the System is SYSTEM_EXTRACT or ECU_EXTRACT the `ecuExtractVersion` attribute shall be defined. ⌋

3 Topology

This chapter explains how a vehicle's physical System Topology is being modeled in AUTOSAR (Example: Figure 3.1). A topology is formed by a number of EcuInstances that are interconnected to each other in order to form ensembles of ECUs and CommunicationClusters, which are further detailed by providing information on bus-specific properties.

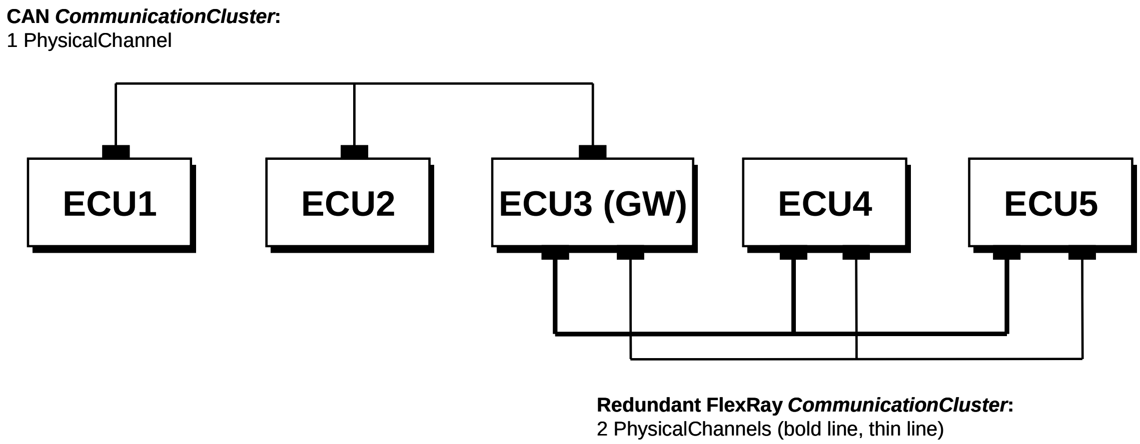

Figure 3.1: Example for a Communication Cluster within a physical network topology

In the AUTOSAR methodology [4] the topology description is one of the inputs for the System Generator. It serves as constraints for mapping the Software Components (see chapter 5.1) contained in the RootSwCompositionPrototype as well as for defining the System Communication matrix (see chapter 6). Gateways which allow the exchange of Signals between CommunicationClusters are covered in chapter 7.

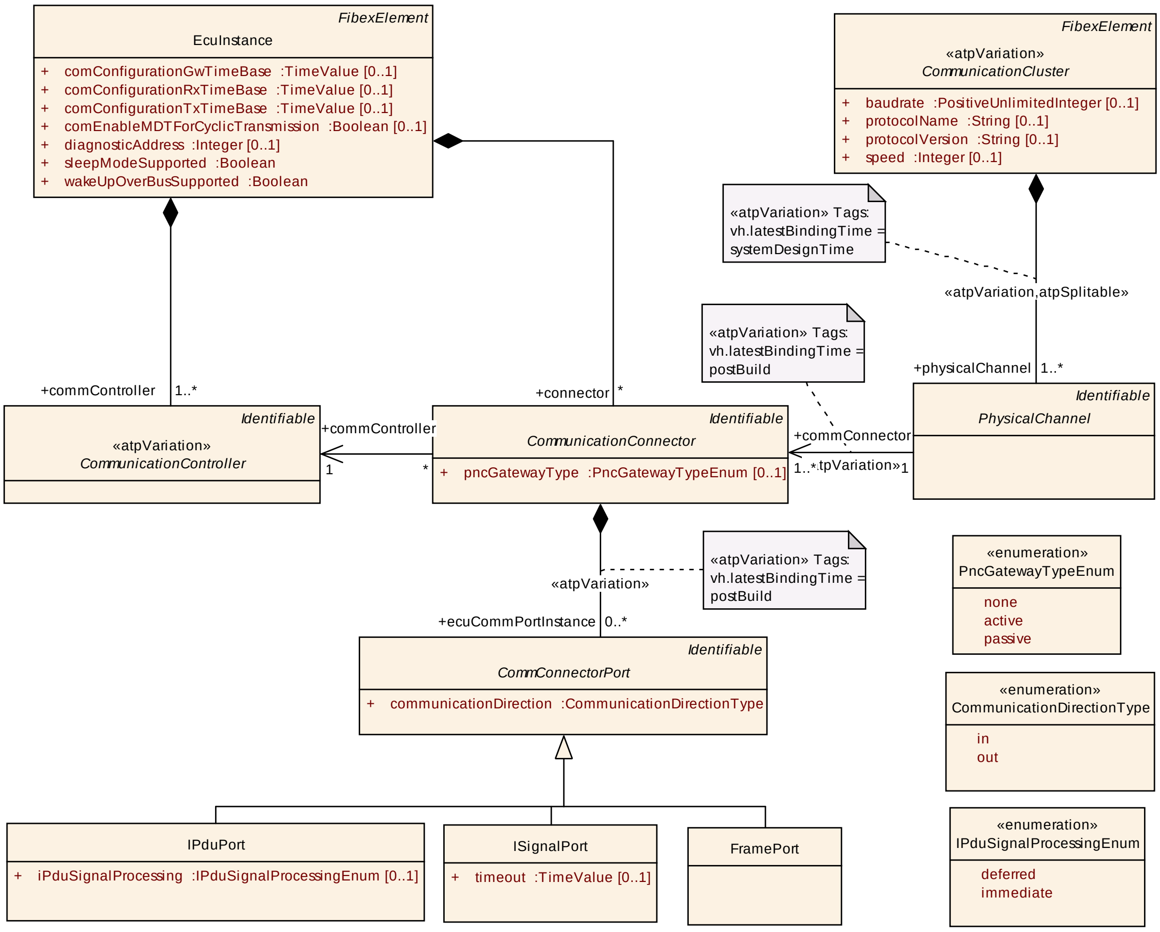

Figure 3.2: Topology elements (Topology)

3.1 ECUs and their communication capabilities

Within a System Topology, the ECUs actually being connected with each other are described in the form of EcuInstances. An EcuInstance needs to have one or more CommunicationController, the actual hardware device by means of which devices send and receive frames from the communication medium. Furthermore, the EcuInstance has one or more CommunicationConnectors which describe the bus interfaces of the ECUs and to specify the sending/receiving behavior.

[TPS_SYST_01004] Definition of AUTOSAR ECU ⌈ In the AUTOSAR sense an ECU means a microcontroller plus peripherals and the according software/configuration. Therefore, each microcontroller requires its own ECU Configuration. ⌋ *(RS_SYST_00013)*

3.1.1 ECU Instance

[TPS_SYST_01005] Definition of `EcuInstance` ⌈ `EcuInstance` describes the presence of a microcontroller in the vehicle. Within an `EcuInstance` class only those properties are described that are subject to system configuration. ⌋*(RS_SYST_00013)*

The actual description of the ECU hardware resources is done by the means of the ECU Resource Template [6]: It uses the `HwElement` class and its aggregated hardware elements for defining a specific ECU type.

[TPS_SYST_01006] Assign `ECU` type to `EcuInstance` ⌈ The process of assigning an `ECU` type to `EcuInstance` is a mapping step (chapter 3.4.1) and performed latest in the System Generation step. ⌋*(RS_SYST_00013)*

An `EcuInstance` can serve as a gateway if it is connected to two or more different clusters by two or more of its `CommunicationController`s.

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,Multilanguage Referrable,PackageableElement,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
associated ComIPduG roup	ISignalIPduGro up	*	ref	With this reference it is possible to identify which ISignalIPduGroups are applicable for which CommunicationConnector/ ECU. Only top level ISignalIPduGroups shall be referenced by an EcuInstance. If an ISignalIPduGroup contains other ISignalIPduGroups than these contained ISignalIPduGroups shall not be referenced by the EcuInstance. Contained ISignalIPduGroups are associated to an EcuInstance via the top level ISignalIPduGroup.
associated PdurIPduG roup	PdurIPduGroup	*	ref	With this reference it is possible to identify which PduR IPdu Groups are applicable for which CommunicationConnector/ ECU.
canTpAddr ess	CanTpAddress	*	ref	Please note that this reference is deprecated and will be removed in future. A Tp Address can be assigned to an ECU without an existing TP Configuration. If TpNodes are described this reference shall not be used. **Tags:** atp.Status=obsolete; atp.StatusRevision Begin=4.1.3
comConfig urationGw TimeBase	TimeValue	0..1	attr	The period between successive calls to Com_MainFunctionRouteSignals of the AUTOSAR COM module in seconds.

Attribute	Datatype	Mul.	Kind	Note
comConfigurationRxTimeBase	TimeValue	0..1	attr	The period between successive calls to Com_MainFunctionRx of the AUTOSAR COM module in seconds.
comConfigurationTxTimeBase	TimeValue	0..1	attr	The period between successive calls to Com_MainFunctionTx of the AUTOSAR COM module in seconds.
comEnableMDTForCyclicTransmission	Boolean	0..1	attr	Enables for the Com module of this EcuInstance the minimum delay time monitoring for cyclic and repeated transmissions (TransmissionModeTiming has cyclicTiming assigned or eventControlledTiming with numberOfRepetitions > 0).
commController	CommunicationController	1..*	aggr	CommunicationControllers of the ECU.
connector	CommunicationConnector	*	aggr	All channels controlled by a single controller.
diagnosticAddress	Integer	0..1	attr	An ECU specific ID for responses of diagnostic routines.
partition	EcuPartition	*	aggr	Optional definition of Partitions within an Ecu.
sleepModeSupported	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.
tpAddress	TpAddress	*	ref	Please note that this reference is deprecated and will be removed in future. A Tp Address can be assigned to an ECU without an existing TP Configuration. If TpNodes are described this reference shall not be used. **Tags:** atp.Status=obsolete; atp.StatusRevisionBegin=4.1.3
wakeUpOverBusSupported	Boolean	1	attr	Driver support for wakeup over Bus.

Table 3.1: 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	***Datatype***	***Mul.***	***Kind***	***Note***
execInUserMode	Boolean	1	attr	A partition can execute either in CPU user mode (execInUserMode = TRUE) or supervisor mode (execInUserMode = 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 3.2: EcuPartition

[constr_3008] EcuInstance subelements ⌈The CommunicationConnector and the CommunicationController that is referenced by the CommunicationConnector must be owned by the same EcuInstance. ⌋

3.1.2 Communication Controller

[TPS_SYST_01007] Definition of CommunicationController ⌈ A CommunicationController is a dedicated hardware device by means of which hosts are sending frames to and receiving frames from the communication medium. ⌋*(RS_SYST_00013)*

[TPS_SYST_01008] Assign CommunicationController to the AUTOSAR Communication Peripheral ⌈ In order to illustrate the relationship of an CommunicationController to the HwElement with category CommunicationController defined in the ECU Resource Description, a mapping between these two classes may be specified using the CommunicationControllerMapping (see chapter 3.4.2). ⌋*(RS_SYST_00013)*

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			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 3.3: CommunicationController

3.1.3 Communication Connector

[TPS_SYST_01009] Definition of CommunicationConnector ⌈ An EcuInstance uses CommunicationConnector elements in order to describe its bus interfaces and to specify the sending/receiving behavior. ⌋*(RS_SYST_00013)*

The relationship between an EcuInstance, a CommunicationController, and a PhysicalChannel is expressed by letting a PhysicalChannel reference a CommunicationConnector (which in turn is aggregated by EcuInstance) and which also has the ability to reference a CommunicationController.

Class	**CommunicationConnector (abstract)**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology			
Note	The connection between the referencing ECU and the referenced channel via the referenced controller. Connectors are used to describe the bus interfaces of the ECUs and to specify the sending/receiving behavior. Each CommunicationConnector has a reference to exactly one communicationController. Note: Several CommunicationConnectors can be assigned to one PhysicalChannel in the scope of one ECU Instance.			
Base	ARObject,Identifiable,MultilanguageReferrable,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
commController	CommunicationController	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.
ecuCommPortInstance	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
pncGatewayType	PncGatewayTypeEnum	0..1	attr	Defines if this EcuInstance shall implement the PncGateway functionality on this CommunicationConnector and its respective PhysicalChannel. Several EcuInstances on the same PhysicalChannel can have the PncGateway functionality enabled, but only one of them shall have the pncGatewayType "active".

Table 3.4: CommunicationConnector

Enumeration	**PncGatewayTypeEnum**
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology
Note	Defines the PncGateway roles.
Literal	*Description*
active	The active PncGateway functionality shall be performed
none	No PncGateway functionality shall be performed
passive	The passive PncGateway functionality shall be performed

Table 3.5: PncGatewayTypeEnum

Note: Use-case for the relation of several CommunicationConnectors assigned to one PhysicalChannel in the scope of one ECUInstance: One safety measure for a safety relevant ECU can be to have two transceivers (and two controllers) connected to the same network (Bus). In case a safety violation is detected one transceiver can be disabled and the respective Frames are blocked. The other transceiver stays active and keeps the ECU alive for diagnostics.

3.2 Communication Clustering

3.2.1 Communication Cluster

[TPS_SYST_01010] Definition of CommunicationCluster ⌈ CommunicationCluster represents a formal way to express that a number of EcuInstances are linked by an arbitrary topology (bus, star, ring, tree). Depending on the communication standard, a CommunicationCluster may either have exactly one or more (redundant) PhysicalChannels. ⌋*(RS_SYST_00013)*

Note that all ECUs within a CommunicationCluster communicate within the same address range.

Note that the same ECU can participate in more than one CommunicationCluster if it has more than one CommunicationConnector being referenced by PhysicalChannels owned by different CommunicationClusters.

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,Multilanguage Referrable,PackageableElement,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
baudrate	PositiveUnlimitedInteger	0..1	attr	Channels speed in bits/s.
physicalChannel	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.
speed	Integer	0..1	attr	This attribute is deprecated and is replaced by the attribute "baudrate". Old description: Channels speed in kbps. **Tags:** atp.Status=obsolete

Table 3.6: CommunicationCluster

Some communication clusters need, additional to the general attributes which are valid for all communication clusters, specialized attributes to describe the individual communication cluster properties. The bustype-specific specializations of `CommunicationCluster` (Figure 3.3) are further detailed in chapter 3.3.

3.2.2 Physical Channel

[TPS_SYST_01011] Definition of `PhysicalChannel` ⌈ `PhysicalChannel` represents the communication medium that is used to send and receive information between communicating ECUs. Each CommunicationCluster has at least one PhysicalChannel. ⌋(RS_SYST_00013)

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			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
commConnector	CommunicationConnector	1..*	ref	Reference to the ECUInstance via a CommunicationConnector 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, variation Point.shortLabel vh.latestBindingTime=postBuild
iSignalTriggering	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, variation Point.shortLabel vh.latestBindingTime=postBuild

Attribute	Datatype	Mul.	Kind	Note
pduTrigger ing	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, variation Point.shortLabel vh.latestBindingTime=postBuild

Table 3.7: PhysicalChannel

3.3 Specialized Attributes of the Topology Entities

According to their characteristic features, different communication standards like FlexRay, CAN, TTCAN, LIN, J1939 and Ethernet have individual attributes that need to be described additionally to the common topology classes. Figure 3.3 shows the specialization of the `CommunicationCluster` into the more specific `FlexrayCluster`, `CanCluster`, `TtcanCluster`, `J1939Cluster`, `LinCluster` and `EthernetCluster`.

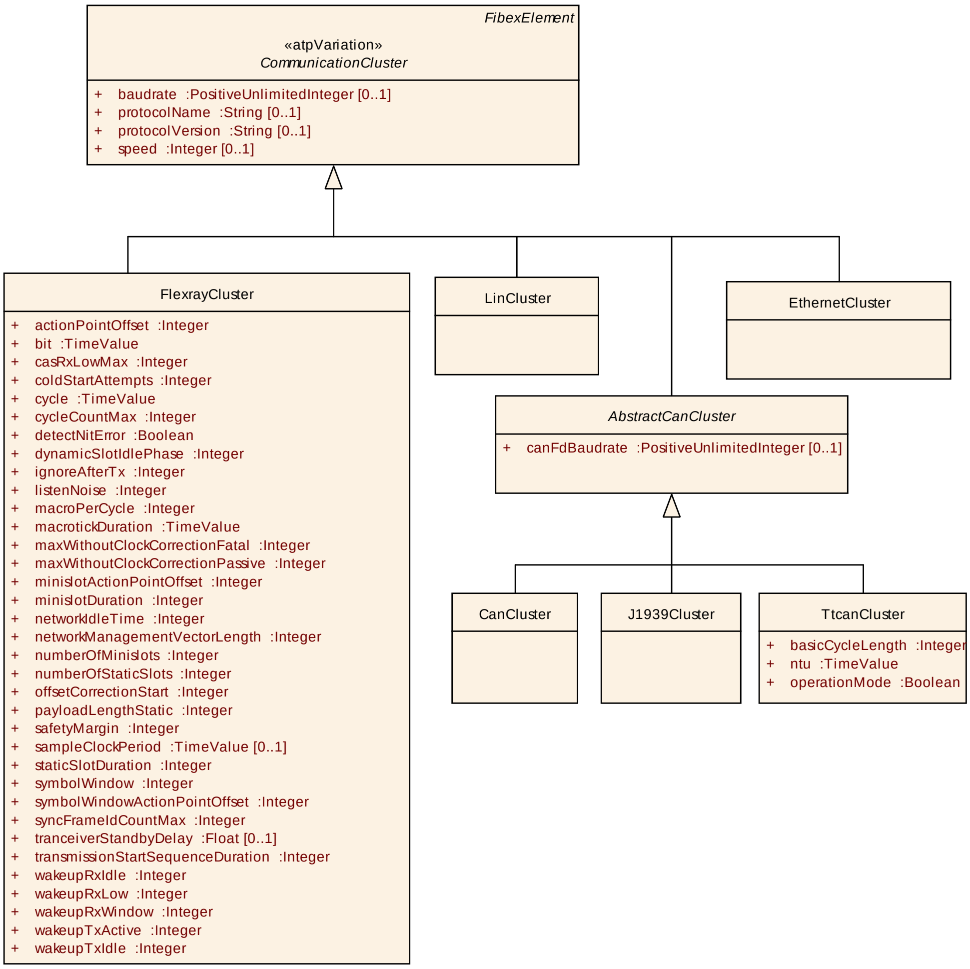

Figure 3.3: Specialized `CommunicationCluster` attributes (TopologyAttributeRefinement)

3.3.1 CAN

Modeling of the CAN bus is supported in the System Template by the means of four specialized meta-model classes: CanCluster, CanCommunicationController, CanPhysicalChannel, CanCommunicationConnector (Figure 3.4).

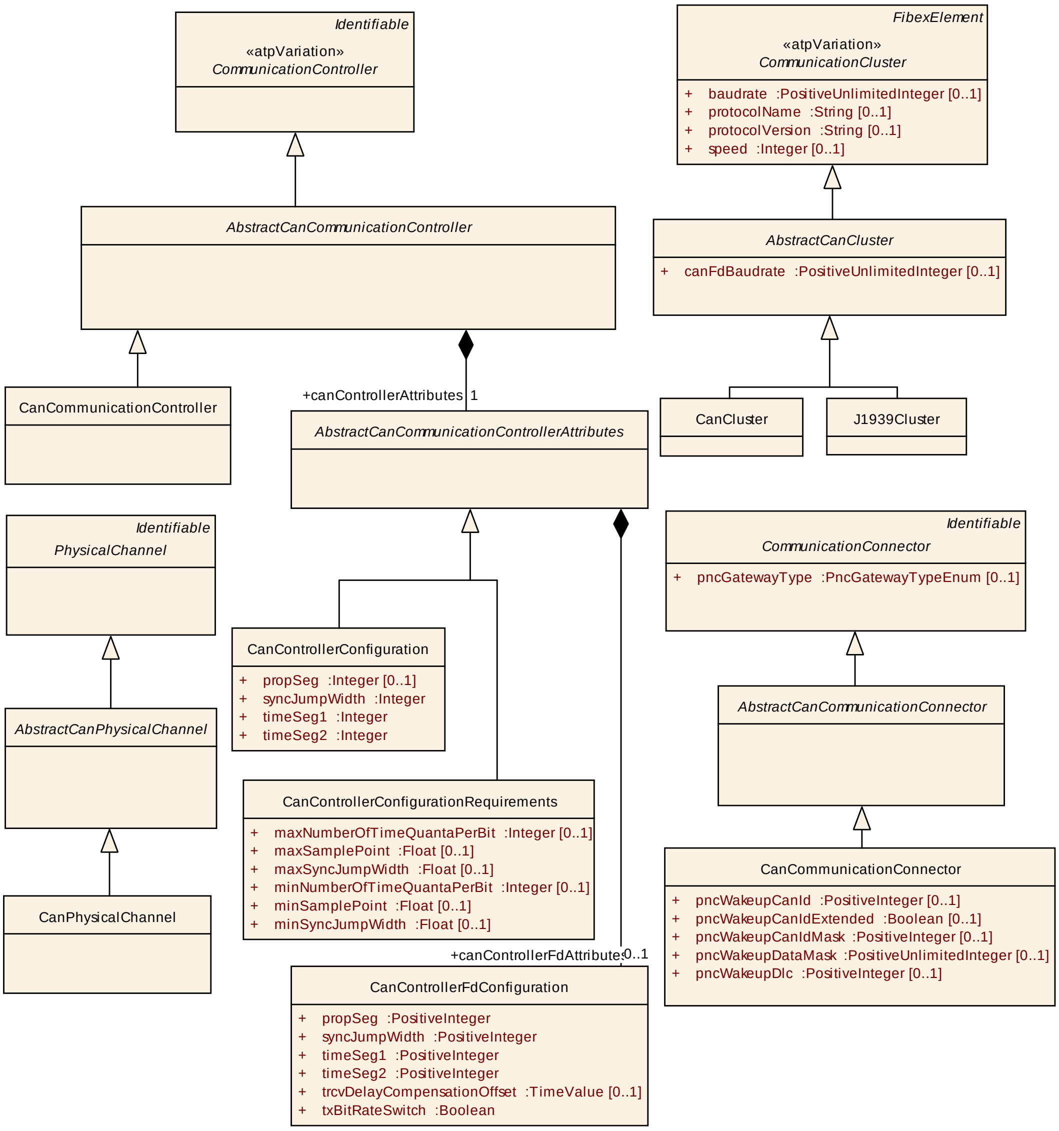

Figure 3.4: CAN bus elements (Fibex4Can_Topology)

3.3.1.1 CAN Cluster

CanCluster specifies the existence of a CAN cluster in the system's physical topology. It contains additional CAN-specific cluster-wide attributes. The common CAN and TTCAN attributes are collected in the AbstractCanCluster class.

Class	≪atpVariation≫ AbstractCanCluster (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	Abstract class that is used to collect the common TtCAN, J1939 and CAN Cluster attributes.			
Base	ARObject,CollectableElement,CommunicationCluster,Fibex Element,Identifiable,MultilanguageReferrable,PackageableElement,Referrable			
Attribute	Datatype	Mul.	Kind	Note
canFdBaudrate	PositiveUnlimitedInteger	0..1	attr	Specifies the data segment baud rate of the controller in bits/s.

Table 3.8: AbstractCanCluster

Class	≪atpVariation≫ CanCluster			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	CAN bus specific cluster attributes. **Tags:** atp.recommendedPackage=CommunicationClusters			
Base	ARObject,AbstractCanCluster,CollectableElement,CommunicationCluster,Fibex Element,Identifiable,MultilanguageReferrable,PackageableElement,Referrable			
Attribute	Datatype	Mul.	Kind	Note
–	–	–	–	–

Table 3.9: CanCluster

3.3.1.2 CAN Communication Controller

`CanCommunicationController` is a specialization of the abstract `CommunicationController` class. It contains the specific CAN controller attributes needed for configuring the CAN stack in an ECU connected to a certain CAN cluster. The common CAN and TTCAN attributes are collected in the `AbstractCanCommunicationController` class. It is possible to specify the CAN Controller configuration parameters as exact values or as requirements that have to be respected by the ECU developer. Therefore the two elements `CanControllerConfiguration` and `CanControllerConfigurationRequirements` were created.

Class	≪atpVariation≫ CanCommunicationController			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	CAN bus specific communication port attributes.			
Base	ARObject,AbstractCanCommunicationController,Communication Controller,Identifiable,MultilanguageReferrable,Referrable			
Attribute	Datatype	Mul.	Kind	Note
–	–	–	–	–

Table 3.10: CanCommunicationController

Class	≪atpVariation≫ **AbstractCanCommunicationController (abstract)**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	Abstract class that is used to collect the common TtCAN and CAN Controller attributes.			
Base	ARObject,CommunicationController,Identifiable,MultilanguageReferrable,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
canControllerAttributes	AbstractCanCommunicationControllerAttributes	1	aggr	CAN Bit Timing configuration

Table 3.11: AbstractCanCommunicationController

Class	**AbstractCanCommunicationControllerAttributes (abstract)**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	For the configuration of the CanController parameters two different approaches can be used: 1. Providing exact values which are taken by the ECU developer (CanControllerConfiguration). 2. Providing ranges of values which are taken as requirements and have to be respected by the ECU developer (CanControllerConfigurationRequirements).			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
canControllerFdAttributes	CanControllerFdConfiguration	0..1	aggr	Bit timing related configuration of a CAN controller for payload and CRC of a CAN FD frame. If this element exists the controller supports CAN FD frames.

Table 3.12: AbstractCanCommunicationControllerAttributes

Class	**CanControllerConfiguration**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	This element is used for the specification of the exact CAN Bit Timing configuration parameter values.			
Base	ARObject,AbstractCanCommunicationControllerAttributes			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
propSeg	Integer	0..1	attr	Specifies propagation delay in time quantas.
syncJumpWidth	Integer	1	attr	The number of quanta in the Synchronization Jump Width, SJW. The (Re-)Synchronization Jump Width (SJW) defines how far a resynchronization may move the Sample Point inside the limits defined by the Phase Buffer Segments to compensate for edge phase errors.
timeSeg1	Integer	1	attr	Specifies phase segment 1 in time quantas. timeSeg1 = Phase_Seg1
timeSeg2	Integer	1	attr	Specifies phase segment 2 in time quantas. timeSeg2 = Phase_Seg2

Table 3.13: CanControllerConfiguration

Class	CanControllerConfigurationRequirements			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	This element allows the specification of ranges for the CAN Bit Timing configuration parameters. These ranges are taken as requirements and have to be respected by the ECU developer.			
Base	ARObject,AbstractCanCommunicationControllerAttributes			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
maxNumberOfTimeQuantaPerBit	Integer	0..1	attr	Maximum number of time quanta in the bit time.
maxSamplePoint	Float	0..1	attr	The max. value of the sample point as a percentage of the total bit time.
maxSyncJumpWidth	Float	0..1	attr	The max. Synchronization Jump Width value as a percentage of the total bit time. The (Re-)Synchronization Jump Width (SJW) defines how far a resynchronization may move the Sample Point inside the limits defined by the Phase Buffer Segments to compensate for edge phase errors.
minNumberOfTimeQuantaPerBit	Integer	0..1	attr	Minimum number of time quanta in the bit time.
minSamplePoint	Float	0..1	attr	The min. value of the sample point as a percentage of the total bit time.
minSyncJumpWidth	Float	0..1	attr	The min. Synchronization Jump Width value as a percentage of the total bit time. The (Re-)Synchronization Jump Width (SJW) defines how far a resynchronization may move the Sample Point inside the limits defined by the Phase Buffer Segments to compensate for edge phase errors.

Table 3.14: CanControllerConfigurationRequirements

[TPS_SYST_01154] CAN Controller support of CAN FD frames ⌈ The bit timing configuration of CAN controllers for CAN FD frames is supported by the `CanControllerFdConfiguration` element that is aggregated by `AbstractCanCommunicationControllerAttributes`. ⌋*(RS_SYST_00048)*

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	**Datatype**	**Mul.**	**Kind**	**Note**
propSeg	PositiveInteger	1	attr	Specifies propagation delay in time quantas.
syncJumpWidth	PositiveInteger	1	attr	Specifies the synchronization jump width for the controller in time quantas.
timeSeg1	PositiveInteger	1	attr	Specifies phase segment 1 in time quantas.
timeSeg2	PositiveInteger	1	attr	Specifies phase segment 2 in time quantas.

Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
trcvDelayCompensationOffset	TimeValue	0..1	attr	Specifies the Transceiver Delay Compensation Offset in seconds. If not specified Transceiver Delay Compensation is disabled.
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 3.15: CanControllerFdConfiguration

3.3.1.3 CAN Physical Channel

CanPhysicalChannel is a specialization of the abstract PhysicalChannel class. It contains the specific CAN PhysicalChannel attributes. The common CAN and TTCAN attributes are collected in the AbstractCanPhysicalChannel class.

Class	**AbstractCanPhysicalChannel (abstract)**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	Abstract class that is used to collect the common TtCAN and CAN PhysicalChannel attributes.			
Base	ARObject,Identifiable,MultilanguageReferrable,PhysicalChannel,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 3.16: AbstractCanPhysicalChannel

Class	**CanPhysicalChannel**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	CAN bus specific physical channel attributes.			
Base	ARObject,AbstractCanPhysicalChannel,Identifiable,MultilanguageReferrable,Physical Channel,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 3.17: CanPhysicalChannel

[constr_3003] Number of CAN channels ⌈CAN clusters shall aggregate exactly one PhysicalChannel.⌋

3.3.1.4 CAN Communication Connector

CanCommunicationConnector is a specialization of the abstract CommunicationConnector class. It contains the specific CAN CommunicationConnector attributes. The common CAN and TTCAN attributes are collected in the AbstractCanCommunicationConnector class.

Class	**AbstractCanCommunicationConnector (abstract)**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	Abstract class that is used to collect the common TtCAN and CAN CommunicationConnector attributes.			
Base	ARObject,CommunicationConnector,Identifiable,MultilanguageReferrable,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 3.18: AbstractCanCommunicationConnector

Class	**CanCommunicationConnector**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	CAN bus specific communication connector attributes.			
Base	ARObject,AbstractCanCommunicationConnector,Communication Connector,Identifiable,MultilanguageReferrable,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
pncWakeupCanId	PositiveInteger	0..1	attr	CAN Identifier used to configure the CAN Transceiver for partial network wakeup.
pncWakeupCanIdExtended	Boolean	0..1	attr	Defines whether pncWakeupCanId and pncWakeupCanIdMask shall be interpreted as extended or standard CAN ID.
pncWakeupCanIdMask	PositiveInteger	0..1	attr	Bit mask for CAN Identifier used to configure the CAN Transceiver for partial network wakeup.
pncWakeupDataMask	PositiveUnlimitedInteger	0..1	attr	Bit mask for CAN Payload used to configure the CAN Transceiver for partial network wakeup. This attribute should not be computed from the pncIdentifier values in order to support future introduction of additional PNCs. Note that for one EcuInstance all contributing pncWakeupDataMask will be bitwise ORed to obtain the value of CanNmPnFilterMaskByte. Note that this data mask is calculated over the whole payload (8 Byte) of the NmPdu ignoring the leading bytes which do not contain pncVector information. The number of leading bytes which shall be ignored is equivalent to the value of System.pncVectorOffset.
pncWakeupDlc	PositiveInteger	0..1	attr	Data Length of the remote data frame used to configure the CAN Transceiver for partial network wakeup in Bytes.

Table 3.19: CanCommunicationConnector

Example: For pncWakeupDataMask = 2^{63} and pncVectorOffset = 2, pncIdentifier with number 63 in a NmPdu will be masked (see Figure 3.5).

NmPdu	Byte 0								Byte 1								Byte 2								Byte 3							
Absolute bit position	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	23	22	21	20	19	18	17	16	31	30	29	28	27	26	25	24
PNC identifiers				N	O	T					U	S	E	D			23	22	21	20	19	18	17	16	31	30	29	28	27	26	25	24

NmPdu	Byte 4								Byte 5								Byte 6								Byte 7							
Absolute bit position	39	38	37	36	35	34	33	32	47	46	45	44	43	42	41	40	55	54	53	52	51	50	49	48	63	62	61	60	59	58	57	56
PNC identifiers	39	38	37	36	35	34	33	32	47	46	45	44	43	42	41	40	55	54	53	52	51	50	49	48	63	62	61	60	59	58	57	56

Figure 3.5: Example of masked pncIdentifiers in a NmPdu

3.3.2 TTCAN

Modeling of TTCAN clusters is supported in the System Template by the means of four specialized meta-model classes: `TtcanCluster`, `TtcanCommunication-`

Controller, TtcanCommunicationConnector, TtcanPhysicalChannel (figure 3.6).

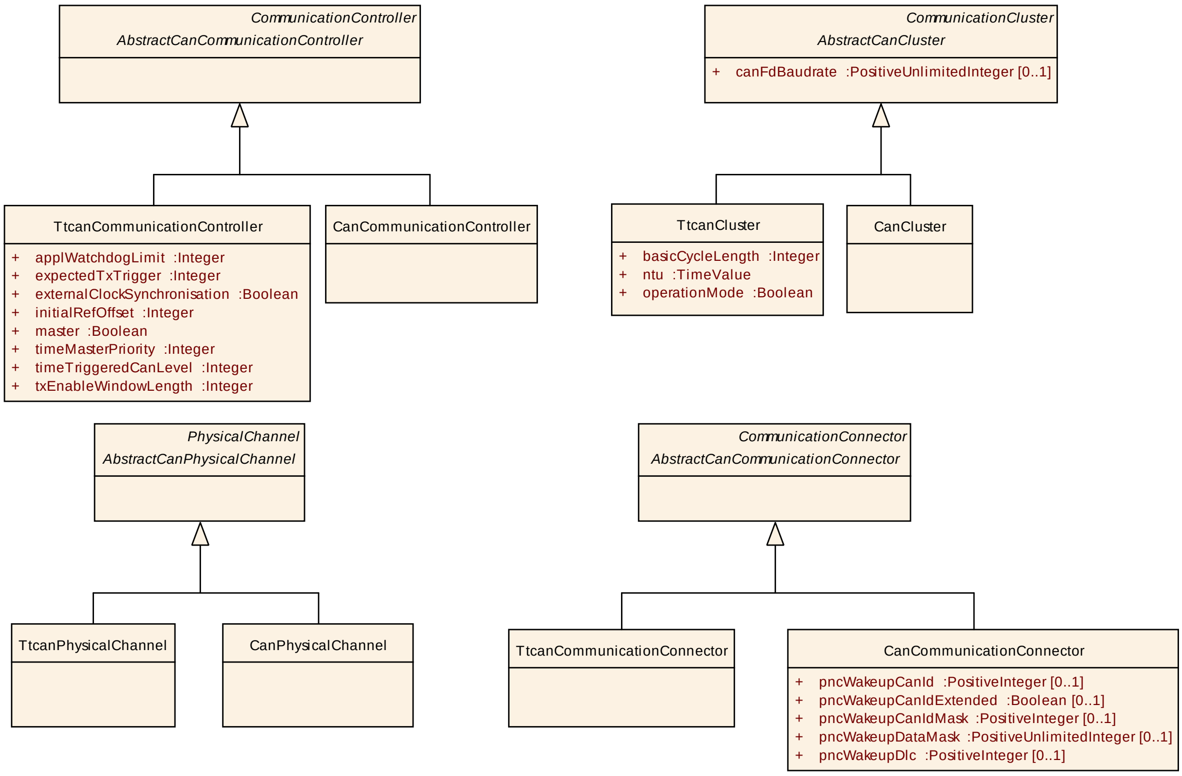

Figure 3.6: TTCAN bus elements (Fibex4Ttcan_Topology)

3.3.2.1 TTCAN Cluster

TtcanCluster specifies the existence of a TTCAN cluster in the system's physical topology. Additionally to the common CAN and TTCAN attributes it contains TTCAN-specific cluster-wide attributes.

Class	≪atpVariation≫ TtcanCluster			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ttcan::TtcanTopology			
Note	TTCAN bus specific cluster attributes. **Tags:** atp.recommendedPackage=CommunicationClusters			
Base	ARObject,AbstractCanCluster,CollectableElement,CommunicationCluster,FibexElement,Identifiable,MultilanguageReferrable,PackageableElement,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
basicCycleLength	Integer	1	attr	Length of a basic-cycle. Unit: NTUs
ntu	TimeValue	1	attr	Unit measuring all times and providing a constant of the whole network. For level 1, this is always the CAN bit time. Unit: seconds.
operationMode	Boolean	1	attr	Possible operation modes True: Time-Triggered False: Event-Synchronised-Time-Triggered

Table 3.20: TtcanCluster

3.3.2.2 TTCAN Communication Controller

TtcanCommunicationController is a specialization of the AbstractCanCommunicationController class. Additionally to the common CAN and TTCAN attributes it contains the specific TTCAN Controller attributes.

Class	≪atpVariation≫ TtcanCommunicationController			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ttcan::TtcanTopology			
Note	TTCAN bus specific communication port attributes.			
Base	ARObject,AbstractCanCommunicationController,CommunicationController,Identifiable,MultilanguageReferrable,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
applWatchdogLimit	Integer	1	attr	The Appl_Watchdog_Limit shall be an 8-bit value specifying the period for the application watchdog in Appl_Watchdog_Limit times 256 NTUs.
expectedTxTrigger	Integer	1	attr	The Expected_Tx_Trigger shall be an eight (8) bit value which limits the number of messages the FSE may try to transmit in one matrix cycle.
externalClockSynchronisation	Boolean	1	attr	One bit shall be used to configure whether or not external clock synchronisation will be allowed during runtime (only Level 2).
initialRefOffset	Integer	1	attr	The Initial_Ref_Offset shall be an eight (8) bit value for the initialisation of Ref_Trigger_Offset.
master	Boolean	1	attr	One bit shall be used to distinguish between (potential) time masters and time slaves. This can be derived from the frame-triggering's triggers.

Attribute	Datatype	Mul.	Kind	Note
timeMasterPriority	Integer	1	attr	The time master priority shall contain a three bit value for the priority of the current time master (the last three bits of the identifier of the reference message). This can be derived from the frame-triggering's triggers.
timeTriggeredCanLevel	Integer	1	attr	One bit shall be used to distinguish between Level 1 and Level 2.
txEnableWindowLength	Integer	1	attr	The length of the Tx_Enable window shall be a four (4) bit value specifying the length of the time period (1-16 nominal CAN bit times) in which a transmission may be started.

Table 3.21: TtcanCommunicationController

3.3.2.3 TTCAN Physical Channel

TtcanPhysicalChannel is a specialization of the AbstractCanPhysicalChannel class. Additionally to the common CAN and TTCAN attributes it contains the specific TTCAN Physical Channel attributes.

Class	TtcanPhysicalChannel			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ttcan::TtcanTopology			
Note	TTCAN bus specific physical channel attributes.			
Base	ARObject,AbstractCanPhysicalChannel,Identifiable,MultilanguageReferrable,PhysicalChannel,Referrable			
Attribute	Datatype	Mul.	Kind	Note
–	–	–	–	–

Table 3.22: TtcanPhysicalChannel

3.3.2.4 TTCAN Communication Connector

TtcanCommunicationConnector is a specialization of the AbstractCanCommunicationConnector class. Additionally to the common CAN and TTCAN attributes it contains the specific TTCAN CommunicationConnector attributes.

Class	TtcanCommunicationConnector			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ttcan::TtcanTopology			
Note	TTCAN bus specific communication connector attributes.			
Base	ARObject,AbstractCanCommunicationConnector,CommunicationConnector,Identifiable,MultilanguageReferrable,Referrable			
Attribute	Datatype	Mul.	Kind	Note
–	–	–	–	–

Table 3.23: TtcanCommunicationConnector

3.3.3 SAE J1939

Modeling of J1939 Communication Clusters is supported in the System Template with the J1939Cluster element that is derived from AbstractCanCluster (see figure 3.4).

Class	≪atpVariation≫ **J1939Cluster**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	J1939 specific cluster attributes. **Tags:** atp.recommendedPackage=CommunicationClusters			
Base	ARObject,AbstractCanCluster,CollectableElement,CommunicationCluster,Fibex Element,Identifiable,MultilanguageReferrable,PackageableElement,Referrable			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
–	–	–	–	–

Table 3.24: J1939Cluster

To describe the communication on a J1939Cluster CanFrameTriggerings are used that are aggregated by a CanPhysicalChannel.

[constr_3050] J1939Cluster uses exactly one CanPhysicalChannel ⌈ A J1939Cluster shall aggregate exactly one CanPhysicalChannel. ⌋

3.3.4 FlexRay

Modeling of FlexRay clusters is supported in the System Template by the means of four specialized meta-model classes: FlexrayCluster, FlexrayCommunicationConnector, FlexrayPhysicalChannel, FlexrayCommunicationController (Figure 3.7).

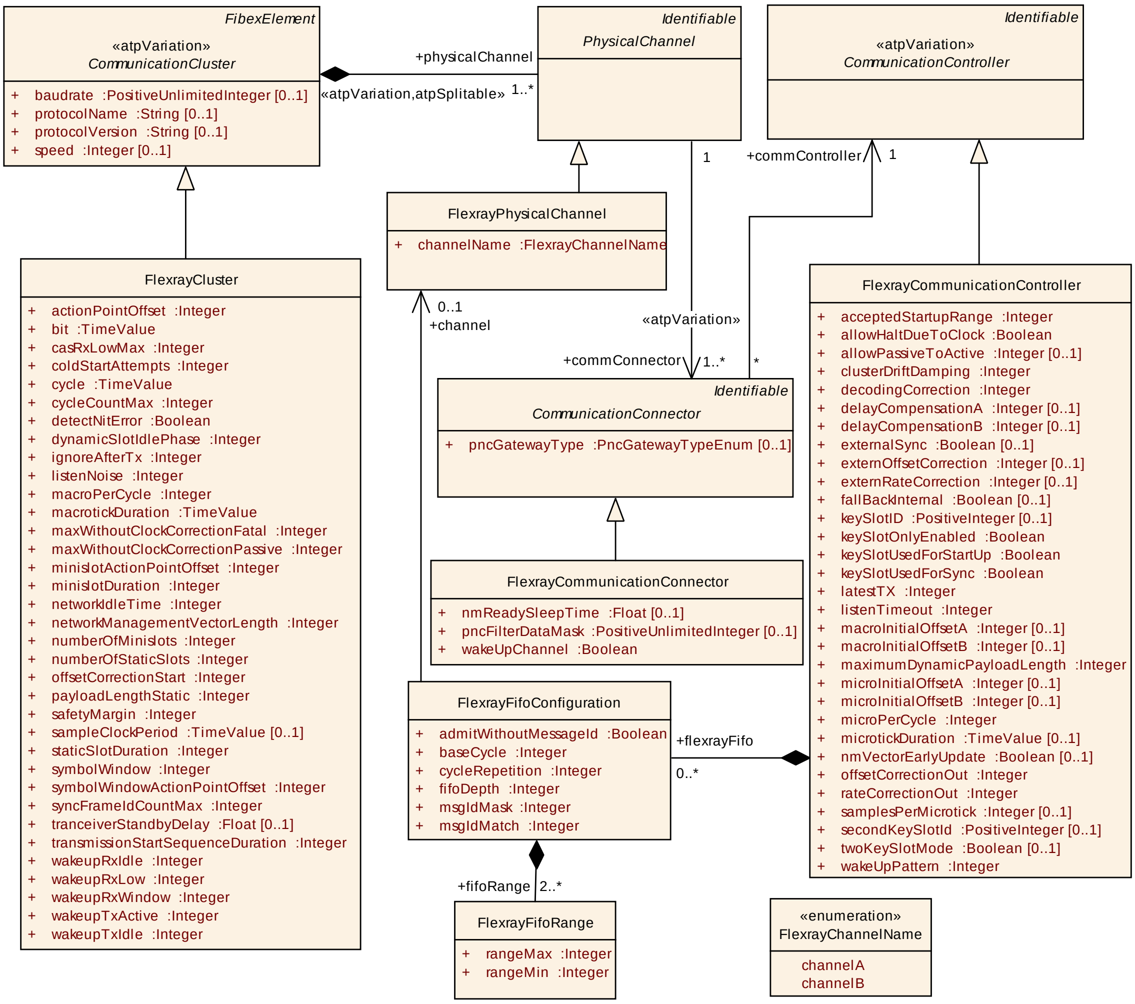

Figure 3.7: FlexRay cluster elements (Fibex4FlexRay_Topology)

3.3.4.1 FlexRay Cluster

FlexrayCluster specifies the existence of a FlexRay cluster in the system's physical topology. It contains additional FlexRay-specific cluster-wide attributes.

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,Fibex Element,Identifiable,MultilanguageReferrable,PackageableElement,Referrable			
Attribute	**Datatype**	**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 = cSamplesPerBit * gdSampleClockPeriod. Unit: seconds (gdBit)
casRxLow Max	Integer	1	attr	Upper limit of the Collision Avoidance Symbol (CAS) acceptance window. Unit:bitDuration
coldStartAt tempts	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
cycleCount Max	Integer	1	attr	Maximum cycle counter value in a given cluster. Remark: Set to 63 for FlexRay Protocol 2.1 Rev. A compliance.
detectNitEr ror	Boolean	1	attr	Indicates whether NIT error status of each cluster shall be detected or not.
dynamicSl otIdlePhas e	Integer	1	attr	The duration of the dynamic slot idle phase in minislots.
ignoreAfter Tx	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
macroPer Cycle	Integer	1	attr	The number of macroticks in a communication cycle
macrotickD uration	TimeValue	1	attr	Duration of the cluster wide nominal macrotick, expressed in s.
maxWithou tClockCorr ectionFatal	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.
maxWithou tClockCorr ectionPass ive	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.

Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
minislotActionPointOffset	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.
networkIdleTime	Integer	1	attr	The duration of the network idle time in macroticks
networkManagementVectorLength	Integer	1	attr	Length of the Network Management vector in a cluster [bytes]
numberOfMinislots	Integer	1	attr	Number of Minislots in the dynamic segment.
numberOfStaticSlots	Integer	1	attr	The number of static slots in the static segment.
offsetCorrectionStart	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
payloadLengthStatic	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.
sampleClockPeriod	TimeValue	0..1	attr	Sample clock period. Unit: seconds
staticSlotDuration	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
symbolWindowActionPointOffset	Integer	1	attr	Number of macroticks the action point offset is from the beginning of the symbol window [Macroticks].
syncFrameIdCountMax	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.
tranceiverStandbyDelay	Float	0..1	attr	The duration of timer t_TrcvStdbyDelay in seconds. The granularity of this parameter shall be restricted to full FlexRay 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.
transmissionStartSequenceDuration	Integer	1	attr	Number of bits in the Transmission Start Sequence [gdBits].

Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
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.
wakeupRxWindow	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 3.25: FlexrayCluster

3.3.4.2 FlexRay Communication Controller

`FlexrayCommunicationController` is a specialization of the `CommunicationController` class. It contains the specific FlexRay controller attributes needed for configuring the FlexRay stack in an ECU connected to a certain FlexRay cluster.

Class	≪atpVariation≫ **FlexrayCommunicationController**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology			
Note	FlexRay bus specific communication port attributes.			
Base	ARObject,CommunicationController,Identifiable,MultilanguageReferrable,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
acceptedStartupRange	Integer	1	attr	Expanded range of measured clock deviation allowed for startup frames during integration. Unit:microtick
allowHaltDueToClock	Boolean	1	attr	Boolean flag that controls the transition to the POC:halt state due to a clock synchronization errors. If set to true, the Communication Controller is allowed to transition to POC:halt. If set to false, the Communication Controller will not transition to the POC:halt state but will enter or remain in the normal POC (passive State).
allowPassiveToActive	Integer	0..1	attr	Number of consecutive even/odd cycle pairs that must have valid clock correction terms before the Communication Controller will be allowed to transition from the POC:normal passive state to POC:normal active state. If set to 0, the Communication Controller is not allowed to transition from POC:norm

Attribute	Datatype	Mul.	Kind	Note
clusterDrift Damping	Integer	1	attr	The cluster drift damping factor used in clock synchronization rate correction in microticks
decodingC orrection	Integer	1	attr	Value used by the receiver to calculate the difference between primary time reference point and secondary time reference point. Unit: Microticks (pDecodingCorrection)
delayCom pensationA	Integer	0..1	attr	Value used to compensate for reception delays on channel A Unit: Microticks. This optional parameter shall only be filled out if channel A is used.
delayCom pensationB	Integer	0..1	attr	Value used to compensate for reception delays on channel B. Unit: Microticks. This optional parameter shall only be filled out if channel B is used.
externOffs etCorrectio n	Integer	0..1	attr	Fixed amount added or subtracted to the calculated offset correction term to facilitate external offset correction, expressed in node-local microticks.
externRate Correction	Integer	0..1	attr	Fixed amount added or subtracted to the calculated rate correction term to facilitate external rate correction, expressed in node-local microticks.
externalSy nc	Boolean	0..1	attr	Flag indicating whether the node is externally synchronized (operating as Time Gateway Sink in an TT-E Time Triggered External Sync cluster) or locally synchronized.
fallBackInt ernal	Boolean	0..1	attr	Flag indicating whether a Time Gateway Sink node will switch to local clock operation when synchronization with the Time Gateway Source node is lost (pFallBackInternal = true) or will instead go to POC:ready (pFallBackInternal = false).
flexrayFifo	FlexrayFifoConfi guration	*	aggr	One First In First Out (FIFO) queued receive structure, defining the admittance criteria to the FIFO.
keySlotID	PositiveInteger	0..1	attr	ID of the slot used to transmit the startup frame, sync frame, or designated single slot frame. If the attributes keySlotUsedForStartUp, keySlotUsedForSync, or keySlotOnlyEnabled are set to true the key slot value is mandatory.
keySlotOnl yEnabled	Boolean	1	attr	Flag indicating whether or not the node shall enter key slot only mode following startup.
keySlotUs edForStart Up	Boolean	1	attr	Flag indicating whether the Key Slot is used to transmit a startup frame.
keySlotUs edForSync	Boolean	1	attr	Flag indicating whether the Key Slot is used to transmit a sync frame.
latestTX	Integer	1	attr	The number of the last minislot in which a transmission can start in the dynamic segment for the respective node

Attribute	Datatype	Mul.	Kind	Note
listenTimeout	Integer	1	attr	Value for the startup listen timeout and wakeup listen timeout. Although this is a node local parameter, the real time equivalent of this value should be the same for all nodes in the cluster. Unit: Microticks
macroInitialOffsetA	Integer	0..1	attr	Integer number of macroticks between the static slot boundary and the closest macrotick boundary of the secondary time reference point based on the nominal macrotick duration. (pMacroInitialOffset). This optional parameter shall only be filled out if channel A is used.
macroInitialOffsetB	Integer	0..1	attr	Integer number of macroticks between the static slot boundary and the closest macrotick boundary of the secondary time reference point based on the nominal macrotick duration. (pMacroInitialOffset). This optional parameter shall only be filled out if channel B is used.
maximumDynamicPayloadLength	Integer	1	attr	Maximum payload length for the dynamic channel of a frame in 16 bit WORDS.
microInitialOffsetA	Integer	0..1	attr	Number of microticks between the closest macrotick boundary described by gMacroInitialOffset and the secondary time reference point. The parameter depends on pDelayCompensationA and therefore it has to be set independently for each channel. This optional parameter shall only be filled out if channel A is used.
microInitialOffsetB	Integer	0..1	attr	Number of microticks between the closest macrotick boundary described by gMacroInitialOffset and the secondary time reference point. The parameter depends on pDelayCompensationB and therefore it has to be set independently for each channel. This optional parameter shall only be filled out if channel B is used.
microPerCycle	Integer	1	attr	The nominal number of microticks in a communication cycle
microtickDuration	TimeValue	0..1	attr	Duration of a microtick. This attribute can be derived from samplePerMicrotick and gdSampleClockPeriod. Unit: seconds
nmVectorEarlyUpdate	Boolean	0..1	attr	Flag indicating when the update of the Network Management Vector in the CHI shall take place. If set to false, the update shall take place after the NIT. If set to true, the update shall take place after the end of the static segment.
offsetCorrectionOut	Integer	1	attr	Magnitude of the maximum permissible offset correction value. Unit:microtick (pOffsetCorrectionOut)

Attribute	Datatype	Mul.	Kind	Note
rateCorrectionOut	Integer	1	attr	Magnitude of the maximum permissible rate correction value and the maximum drift offset between two nodes operating with unsynchronized clocks for one communication cycle. Unit:Microticks (pRateCorrectionOut) Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter pdMaxDrift.
samplesPerMicrotick	Integer	0..1	attr	Number of samples per microtick
secondKeySlotId	PositiveInteger	0..1	attr	ID of the second Key slot, in which a second startup frame shall be sent in TT-L Time Triggered Local Master Sync or TT-E Time Triggered External Sync mode. If this parameter is set to zero the node does not have a second key slot.
twoKeySlotMode	Boolean	0..1	attr	Flag indicating whether node operates as a startup node in a TT-E Time Triggered External Sync or TT-L Time Triggered Local Master Sync cluster.
wakeUpPattern	Integer	1	attr	Number of repetitions of the Tx-wakeup symbol to be sent during the CC_WakeupSend state of this Node in the cluster

Table 3.26: FlexrayCommunicationController

Class	**FlexrayFifoConfiguration**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology			
Note	One First In First Out (FIFO) queued receive structure, defining the admittance criteria to the FIFO, and mandating the ability to admit messages into the FIFO based on Message Id filtering criteria.			
Base	ARObject			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
admitWithoutMessageId	Boolean	1	attr	Boolean configuration which determines whether or not frames received in the dynamic segment that don't contain a message ID will be admitted into the FIFO.
baseCycle	Integer	1	attr	FIFO cycle counter acceptance criteria.
channel	FlexrayPhysicalChannel	0..1	ref	Fifo channel admittance criteria.
cycleRepetition	Integer	1	attr	FIFO cycle counter acceptance criteria.
fifoDepth	Integer	1	attr	FrFifoDepth configures the maximum number of rx-frames which can be contained in the FIFO.
fifoRange	FlexrayFifoRange	2..*	aggr	FIFO Frame Id range acceptance criteria.
msgIdMask	Integer	1	attr	FIFO message identifier acceptance criteria (Mask filter).
msgIdMatch	Integer	1	attr	FIFO message identifier acceptance criteria (Match filter).

Table 3.27: FlexrayFifoConfiguration

Class	**FlexrayFifoRange**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology			
Note	FIFO Frame Id range acceptance criteria.			
Base	ARObject			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
rangeMax	Integer	1	attr	Max Range.
rangeMin	Integer	1	attr	Min Range.

Table 3.28: FlexrayFifoRange

3.3.4.3 FlexRay Communication Connector

FlexrayCommunicationConnector adds the FlexRay specific attributes to the CommunicationConnector.

Class	**FlexrayCommunicationConnector**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology			
Note	FlexRay specific attributes to the CommunicationConnector			
Base	ARObject,CommunicationConnector,Identifiable,MultilanguageReferrable,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
nmReadySleepTime	Float	0..1	attr	The value of this attribute influences the shutdown behavior of the FlexRay NM. FrNm switches to bus sleep mode nmReadySleepTime seconds after the completion of the last repetition cycle containing a NM vote.
pncFilterDataMask	PositiveUnlimitedInteger	0..1	attr	Bit mask for FlexRay Payload used to configure the FlexRay Transceiver for partial network wakeup. This attribute should not be computed from the pncIdentifier values in order to support future introduction of additional PNCs. Note that for one EcuInstance all contributing pncFilterDataMask will be bitwise ORed to obtain the value of FrNmPnFilterMaskByte. Note that this data mask is calculated over the whole payload (8 Byte) of the NmPdu ignoring the leading bytes which do not contain pncVector information. The number of leading bytes which shall be ignored is equivalent to the value of System.pncVectorOffset.
wakeUpChannel	Boolean	1	attr	Referenced channel used by the node to send a wakeup pattern. (pWakeupChannel)

Table 3.29: FlexrayCommunicationConnector

[constr_3508] Value of nmReadySleepTime ⌈ The nmReadySleepTime value shall be a multiple of cycle * nmRepetitionCycle. ⌋

The masking of pncIdentifiers in a NmPdu based on the pncWakeupDataMask is done in the same way as for the CanCommunicationConnector (see Example in Figure 3.5).

3.3.4.4 FlexRay Physical Channel

FlexrayPhysicalChannel adds the FlexRay specific attributes to the PhysicalChannel.

Class	**FlexrayPhysicalChannel**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology			
Note	FlexRay specific attributes to the physicalChannel			
Base	ARObject,Identifiable,MultilanguageReferrable,PhysicalChannel,Referrable			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
channelName	FlexrayChannelName	1	attr	Name of the channel (Channel A or Channel B).

Table 3.30: FlexrayPhysicalChannel

Enumeration	**FlexrayChannelName**
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Flexray::Flexray Topology
Note	Name of the channel.
Literal	***Description***
channelA	Channel A
channelB	Channel B

Table 3.31: FlexrayChannelName

[constr_3018] Number of FlexRay channels ⌈A FlexrayCluster shall use either one FlexrayPhysicalChannel with channelName set to either channelA or channelB or else two FlexrayPhysicalChannels with one channelName channelA and one channelName channelB. ⌋

3.3.5 LIN

A LinCluster consists of exactly one master node connected to several slave nodes. The master is responsible for providing the frame headers on the bus according to a predefined schedule, whereas the slaves send or receive the actual frame information ([10]).

[TPS_SYST_01012] Different Properties of LinMaster and LinSlave ⌈ In the System Template the different properties of master and slave nodes are handled by deriving the LIN-specific subclasses LinMaster and LinSlave as specializations of LinCommunicationController. ⌋*(RS_SYST_00022)*

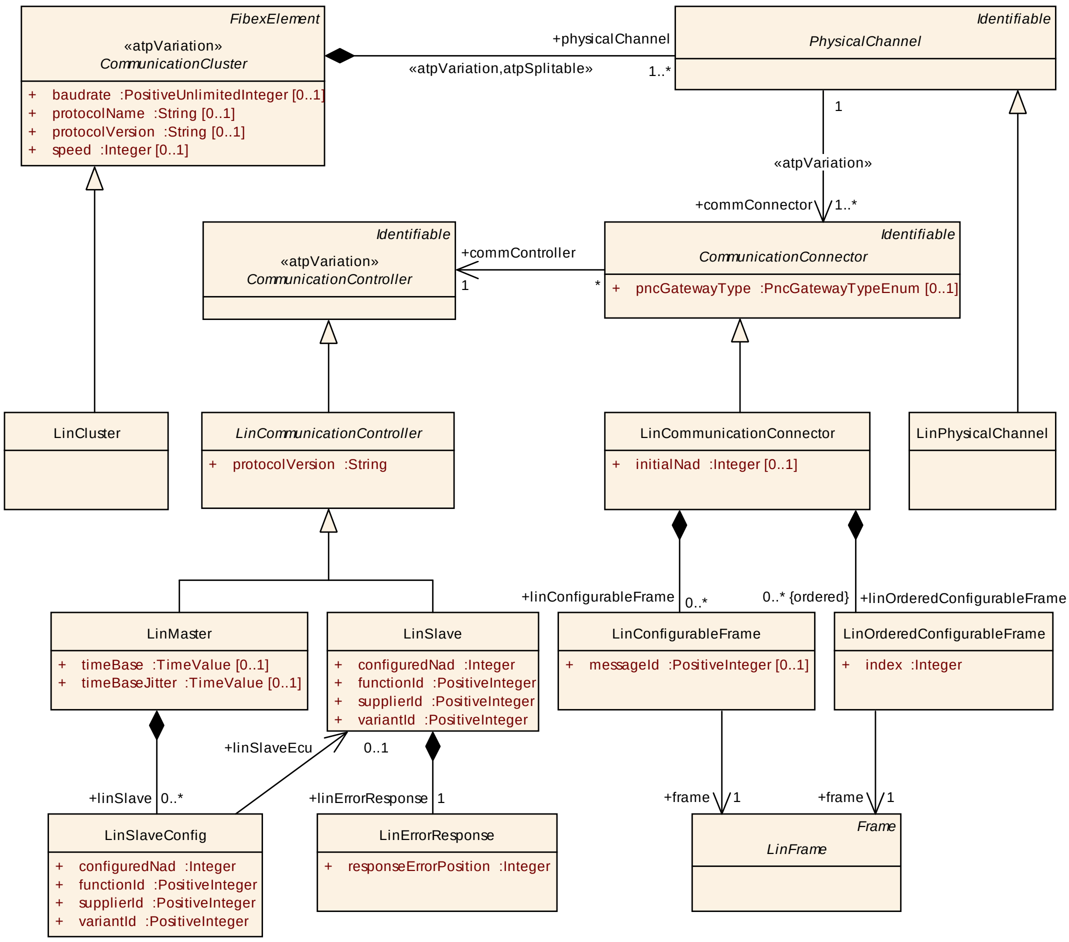

Figure 3.8: Specialized LinCommunicationController attributes (Fibex4Lin_Topology)

Note that the AUTOSAR BSW only supports LIN masters. LIN slaves are seen as non AUTOSAR ECUs. They can be described in the System Template in order to configure the LIN Interface for the master correctly, but AUTOSAR does not support the development of LIN slaves as of AUTOSAR release 4.0 ([14], [15]).

3.3.5.1 LIN Cluster

LinCluster specifies the existence of a LIN cluster in the system's physical topology.

Class	≪atpVariation≫ LinCluster			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinTopology			
Note	LIN specific attributes **Tags:** atp.recommendedPackage=CommunicationClusters			
Base	ARObject,CollectableElement,CommunicationCluster,FibexElement,Identifiable,MultilanguageReferrable,PackageableElement,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
–	–	–	–	–

Table 3.32: LinCluster

3.3.5.2 LIN Communication Controller

LinCommunicationController is a specialization of the CommunicationController class. It is an abstract class, to be further specialized by LinMaster and LinSlave.

Class	≪atpVariation≫ LinCommunicationController (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinTopology			
Note	LIN bus specific communication controller attributes.			
Base	ARObject,CommunicationController,Identifiable,MultilanguageReferrable,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
protocolVersion	String	1	attr	Version specifier for a communication protocol.

Table 3.33: LinCommunicationController

3.3.5.3 LIN Master

LinMaster describes the existence of a LIN master task in a LIN topology node. As such it contains the attributes specific to a LIN master task.

Class	≪atpVariation≫ LinMaster			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinTopology			
Note	Describing the properties of the refering ecu as a LIN master.			
Base	ARObject,CommunicationController,Identifiable,LinCommunicationController,MultilanguageReferrable,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
linSlave	LinSlaveConfig	*	aggr	LinSlaves that are handled by the LinMaster.

Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
timeBase	TimeValue	0..1	attr	Time base is mandatory for the master. It is not used for slaves. LIN 2.0 Spec states: "The time_base value specifies the used time base in the master node to generate the maximum allowed frame transfer time." The time base shall be specified AUTOSAR conform in seconds.
timeBaseJitter	TimeValue	0..1	attr	The attribute timeBaseJitter is a mandatory attribute for the master and not used for slaves. LIN 2.0 Spec states: "The jitter value specifies the differences between the maximum and minimum delay from time base start point to the frame header sending start point (falling edge of BREAK signal)." The jitter shall be specified AUTOSAR conform in seconds.

Table 3.34: LinMaster

Class	**LinSlaveConfig**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinTopology			
Note	Node attributes of LIN slaves that are handled by the LinMaster. In the System Description LIN slaves may be described as non AUTOSAR ECUs (linSlaveEcu reference). But in an Ecu Extract of the LinMaster the LinSlaveEcus will not be available. The information that is described here is necessary in the ECU Extract for the configuration of the LinMaster. The values of attributes of LinSlaveConfig and LinSlave shall be identical for each LinSlaveConfig that points to a LinSlave. Please note that this causes redundancy which is intended to support flexible development methodology.			
Base	ARObject			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
configuredNad	Integer	1	attr	To distinguish LIN slaves that are used twice or more within the same cluster.
functionId	PositiveInteger	1	attr	LIN function ID.
linSlaveEcu	LinSlave	0..1	ref	Reference to the LinSlaveEcu.
supplierId	PositiveInteger	1	attr	LIN Supplier ID.
variantId	PositiveInteger	1	attr	Specifies the Variant ID.

Table 3.35: LinSlaveConfig

[constr_3034] Values of `LinSlaveConfig` and `LinSlave` attributes ⌈ The values of attributes of `LinSlaveConfig` and `LinSlave` shall be identical for each `LinSlaveConfig` that points to a `LinSlave`. ⌋

Please note that this causes redundancy which is intended to support flexible development methodology.

[TPS_SYST_01046] ShortNames of `LinSlaveConfig` and `LinSlave` ⌈ The `shortName`s of a pair of `LinSlaveConfig` and `LinSlave` do not have to be identical. ⌋

3.3.5.4 LIN Slave

`LinSlave` describes the existence of a LIN slave task in a LIN topology node. It describes the attributes of a single LIN slave node. AUTOSAR doesn't support LIN slave functionality in an AUTOSAR ECU, thus not the full FIBEX description of a slave node, but rather the subset of attributes of a Node Capability File (ncf, see [10]) relevant as requirements for configuring the master are included in the System Template.

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,MultilanguageReferrable,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
configuredNad	Integer	1	attr	To distinguish LIN slaves that are used twice or more within the same cluster.
functionId	PositiveInteger	1	attr	LIN function ID
linErrorResponse	LinErrorResponse	1	aggr	Each slave node shall publish one response error in one of its transmitted unconditional frames.
supplierId	PositiveInteger	1	attr	LIN Supplier ID
variantId	PositiveInteger	1	attr	Specifies the Variant ID

Table 3.36: LinSlave

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	**Datatype**	**Mul.**	**Kind**	**Note**
frameTriggering	LinFrameTriggering	1	ref	Reference to an unconditional frame that transmits the response error. The referenced LinFrameTriggering shall contain a reference to an unconditionalFrame.
responseErrorPosition	Integer	1	attr	Specifies the position of the ResponseError bit in the frame. Each slave node shall publish one response error in one of its transmitted unconditional frames.

Table 3.37: LinErrorResponse

3.3.5.5 LIN Communication Connector

LinCommunicationConnector is a specialization of the CommunicationConnector class. The LinCommunicationConnector element contains lists of frames processed by the slave node.

[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. ⌋

Class	**LinCommunicationConnector**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinTopology			
Note	LIN bus specific communication connector attributes.			
Base	ARObject,CommunicationConnector,Identifiable,MultilanguageReferrable,Referrable			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
initialNad	Integer	0..1	attr	Initial NAD of the LIN slave.
linConfigurableFrame	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.
linOrderedConfigurableFrame (ordered)	LinOrderedConfigurableFrame	*	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 3.38: 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	*Datatype*	*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 3.39: LinConfigurableFrame

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	*Datatype*	*Mul.*	*Kind*	*Note*
frame	LinFrame	1	ref	Reference to a Frame that is processed by the slave node.

Attribute	Datatype	Mul.	Kind	Note
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 3.40: LinOrderedConfigurableFrame

3.3.5.6 LIN Physical Channel

LinPhysicalChannel is a specialization of the PhysicalChannel class. It contains additional Lin-specific PhysicalChannel attributes.

Class	LinPhysicalChannel			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinTopology			
Note	LIN specific attributes to the physicalChannel			
Base	ARObject,Identifiable,MultilanguageReferrable,PhysicalChannel,Referrable			
Attribute	Datatype	Mul.	Kind	Note
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 3.41: LinPhysicalChannel

[constr_3015] Number of LIN channels ⌈LIN clusters shall aggregate exactly one LinPhysicalChannel.⌋

3.3.6 Ethernet

To describe the Ethernet at the data link- and physical layer the following System Template meta-model classes are used: `EthernetCluster`, `EthernetCommunicationController`, `EthernetCommunicationConnector`, `EthernetPhysicalChannel`, `CouplingElement`, `CouplingPort` and `CouplingPortConnection` (see Figure 3.9).

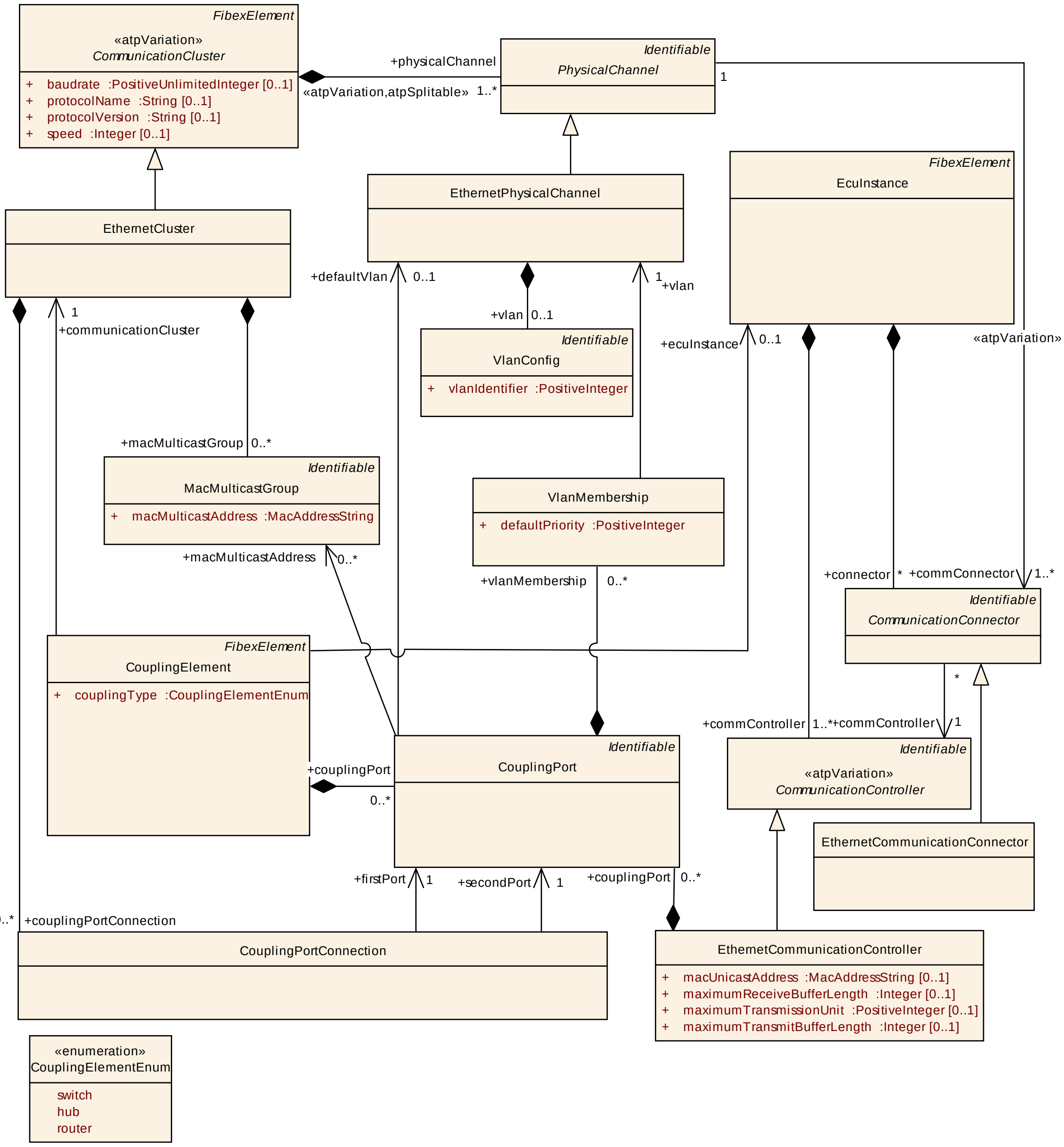

Figure 3.9: Ethernet topology elements (Fibex4Ethernet_Topology)

3.3.6.1 Ethernet Cluster

Each EthernetCluster may have globally defined MacMulticastGroups. MacMulticastGroups have a macMulticastAddress (canonical Format FF:FF:FF:FF:FF:FF). One sender can handle many receivers simultaneously, if the receivers have all the same macMulticastAddress.

[constr_3047] Uniqueness of macMulticastAddresses ⌈ A macMulticastAddress shall be unique in a particular EthernetCluster. ⌋

Class	≪atpVariation≫ **EthernetCluster**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::Ethernet Topology			
Note	Ethernet-specific cluster attributes. **Tags:** atp.recommendedPackage=CommunicationClusters			
Base	ARObject,CollectableElement,CommunicationCluster,Fibex Element,Identifiable,MultilanguageReferrable,PackageableElement,Referrable			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
couplingPortConnection	CouplingPortConnection	*	aggr	Specification of connections between CouplingElements and EcuInstances.
macMulticastGroup	MacMulticastGroup	*	aggr	MacMulticastGroup that is defined for the Subnet (EthernetCluster).

Table 3.42: EthernetCluster

Class	**MacMulticastGroup**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::Ethernet Topology			
Note	Per EthernetCluster globally defined MacMulticastGroup. One sender can handle many receivers simultaneously if the receivers have all the same macMulticastAddress. The addresses need to be unique for the particular EthernetCluster.			
Base	ARObject,Identifiable,MultilanguageReferrable,Referrable			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
macMulticastAddress	MacAddressString	1	attr	A multicast MAC address (Media Access Control address) is a identifier for a group of hosts in a network.

Table 3.43: MacMulticastGroup

3.3.6.2 Ethernet Physical Channel

The EthernetPhysicalChannel represents a VLAN. VLANs (IEEE 802.1q) divide physical Ethernet networks in logical subnets. Their realization requires switches with VLAN support. VLANs are defined on a switch on a port-by-port basis.

[TPS_SYST_01095] tagged VLANs ⌈ In the System Description a VLAN is represented by an EthernetPhysicalChannel and is identified by its vlanIdentifier. ⌋*(RS_SYST_00039)*

[TPS_SYST_01096] untagged VLANs ⌈ If the VlanConfig and the vlanIdentifier are not defined for an EthernetPhysicalChannel than the channel is called "untagged". ⌋*(RS_SYST_00039)*

Every Frame that is sent over a "tagged" VLAN is tagged with a VLAN Tag. With this tag every receiving switch has the information about the VLAN that the Frame belongs to. The VLAN Tag that is attached to a Frame contains the user priority for the Frame that is described with the defaultPriority and the vlanIdentifier.

Class	**EthernetPhysicalChannel**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::Ethernet Topology			
Note	The EthernetPhysicalChannel represents a VLAN or an untagged channel. An untagged channel is modeled as an EthernetPhysicalChannel without an aggregated VLAN.			
Base	ARObject,Identifiable,MultilanguageReferrable,PhysicalChannel,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
networkEndpoint	NetworkEndpoint	*	aggr	Collection of NetworkEndpoints that are used in the VLan.
soAdConfig	SoAdConfig	1	aggr	SoAd Configuration for one specific Physical Channel.
vlan	VlanConfig	0..1	aggr	VLAN Configuration.

Table 3.44: EthernetPhysicalChannel

[TPS_SYST_01086] Number of Ethernet channels ⌈ Each EthernetCluster may aggregate up to 4096 EthernetPhysicalChannels. ⌋*(RS_SYST_00039)*

Class	**VlanConfig**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::Ethernet Topology			
Note	VLAN Configuration attributes			
Base	ARObject,Identifiable,MultilanguageReferrable,Referrable			
Attribute	***Datatype***	***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 3.45: VlanConfig

[constr_3048] Range of vlanIdentifier ⌈ The allowed values of vlanIdentifier range from 0 to 4095. ⌋

3.3.6.3 Ethernet Coupling Elements and Coupling Ports

A CouplingElement is used to connect EcuInstances via CouplingPorts to EthernetPhysicalChannels (VLANs) that are defined within an EthernetCluster.

CouplingElements can reach from a simple hub to a complex managed switch or even devices with functionalities in higher layers. A CouplingElement references the EthernetCluster and contains a collection of available CouplingPorts. The couplingType identifies the CouplingElement as a switch, hub or router.

Class	**CouplingElement**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::Ethernet Topology			
Note	A CouplingElement is used to connect EcuInstances to the VLAN of an EthernetCluster. CouplingElements can reach from a simple hub to a complex managed switch or even devices with functionalities in higher layers. A CouplingElement that is not related to an EcuInstance occurs as a dedicated single device. **Tags:** atp.recommendedPackage=CouplingElements			
Base	ARObject,CollectableElement,FibexElement,Identifiable,Multilanguage Referrable,PackageableElement,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
communicationCluster	EthernetCluster	1	ref	This relationship defines to which cluster the CouplingElement belongs.
couplingPort	CouplingPort	*	aggr	Hardware Port of the CouplingElement that is used to connect this CouplingPort to EcuInstances or other CouplingElements.
couplingType	CouplingElementEnum	1	attr	Describes the coupling type of this CouplingElement.
ecuInstance	EcuInstance	0..1	ref	Optional reference to the ECU where the CouplingElement is located.

Table 3.46: CouplingElement

Enumeration	**CouplingElementEnum**
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::Ethernet Topology
Note	Identifies the Coupling type.
Literal	***Description***
hub	A device that is used to connect segments of a LAN. In Hubs frames are "broadcasted" to every one of its ports.
router	A device that routes frames between different networks.
switch	A device that filters and forwards frames between different LAN segments.

Table 3.47: CouplingElementEnum

[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. ⌋

Class	**CouplingPort**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::Ethernet Topology			
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	***Datatype***	***Mul.***	***Kind***	***Note***
defaultVlan	EthernetPhysicalChannel	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).
macMulticastAddress	MacMulticastGroup	*	ref	Static MAC-Multicast-Address binding to a CouplingPort. This supports the sending of MAC-Multicast-Messages via the CouplingPort.
vlanMembership	VlanMembership	*	aggr	Messages of VLANs that are defined here can be communicated via the CouplingPort.

Table 3.48: CouplingPort

Class	**VlanMembership**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::Ethernet Topology			
Note	Static logical channel or VLAN binding to a switch-port.			
Base	ARObject			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
defaultPriority	PositiveInteger	1	attr	Standard output-priority outgoing Frames will be tagged with. This allows to assign different defaultPriorities to each VLAN.
vlan	EthernetPhysicalChannel	1	ref	References a channel that represents a VLAN or an untagged channel.

Table 3.49: VlanMembership

Class	CouplingPortConnection			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::Ethernet Topology			
Note	Connection between two CouplingPorts (firstPort and secondPort).			
Base	ARObject			
Attribute	**Datatype**	**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 3.50: CouplingPortConnection

CouplingPorts are hardware ports of CouplingElements and EcuInstances. Connections between CouplingPorts are realized through CouplingPortConnections.

Optionally the CouplingPort of a CouplingElement may also have one or several VlanMemberships, a defaultVlan reference and a reference to a MacMulticastGroup.

[TPS_SYST_01097] Assignment of CouplingPorts to a VLAN ⌈ CouplingPorts of CouplingElements can be assigned to VLANs (EthernetPhysicalChannels) with the vlanMembership aggregation. ⌋*(RS_SYST_00039)*

[TPS_SYST_01098] Assignment of CouplingPorts to an "untagged" VLAN ⌈ A CouplingPort may be assigned to several VLANs, but only one of those assignments can be "untagged". ⌋*(RS_SYST_00039)*

Figure 3.10 shows a CouplingElement with two CouplingPorts.

In this example Port 0 is assigned to three VLANs and one "untagged" EthernetPhysicalChannel. VLAN3 is marked as the defaultVlan. With the combination of the defaultVlan and the VlanMembership to the "untagged" EthernetPhysicalChannel the Frames that are transmitted over Port 0 on VLAN3 are "untagged" on the wire in both directions (Tx and Rx). The switch adds the tag for incoming untagged messages at the port (ingress tagging) and for outgoing messages the tag is removed at the port (egress untagging).

Port 1 is assigned to three VLANs. But the VlanMembership to the "untagged" EthernetPhysicalChannel is not defined here. For this reason, Frames that are transmitted over Port 1 on VLAN3 are "tagged".

If a defaultVlan is defined for a CouplingPort but the defaultVlan is not referenced by the VlanMembership then "untagged" Frames can be received via the CouplingPort. But a response can not be send back.

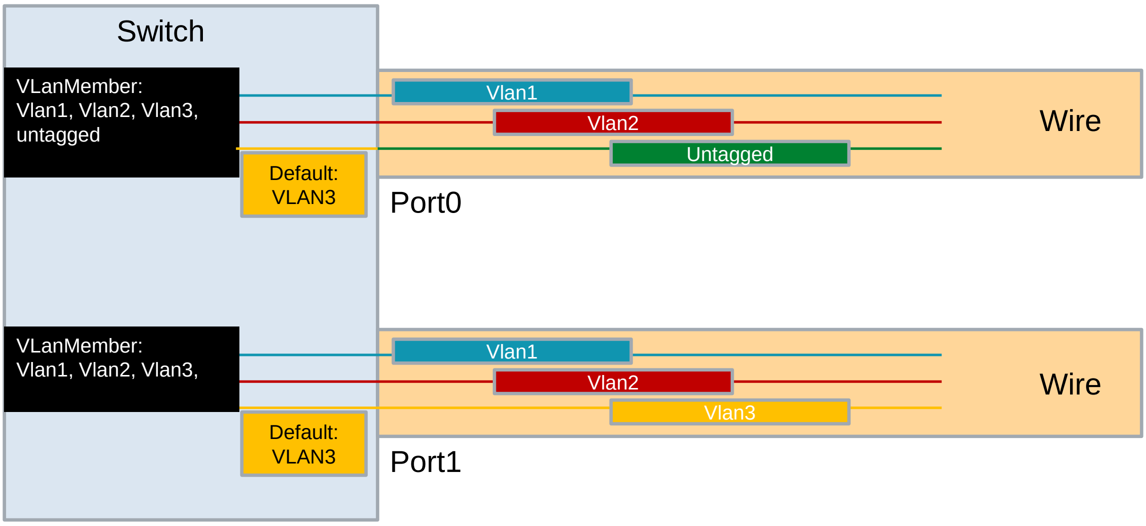

Figure 3.10: Default Vlan Example

3.3.6.4 Ethernet Communication Controller

EthernetCommunicationController is a specialization of the Communication-Controller class. It contains the specific Ethernet controller attributes needed for configuring an EcuInstance connected to a certain Ethernet cluster.

Class	≪atpVariation≫ **EthernetCommunicationController**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::Ethernet Topology			
Note	Ethernet specific communication port attributes.			
Base	ARObject,CommunicationController,Identifiable,MultilanguageReferrable,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
couplingPort	CouplingPort	*	aggr	Optional CouplingPort that can be used to connect the ECU to a CouplingElement (e.g. a switch).
macUnicastAddress	MacAddressString	0..1	attr	Media Access Control address (MAC address) that uniquely identifies each EthernetCommunicationController in the network.
maximumReceiveBufferLength	Integer	0..1	attr	Determines the maximum receive buffer length (frame length) in bytes.
maximumTransmissionUnit	PositiveInteger	0..1	attr	This attribute specifies the maximum transmission unit in bytes.
maximumTransmitBufferLength	Integer	0..1	attr	Determines the maximum transmit buffer length (frame length) in bytes.

Table 3.51: EthernetCommunicationController

The EthernetCommunicationController has the additional information of a macUnicastAddress. This is a globally unique MAC-address for the CommunicationController.

3.3.6.5 Ethernet Communication Connector

EthernetCommunicationConnector adds the Ethernet specific attributes to the CommunicationConnector.

Class	**EthernetCommunicationConnector**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::Ethernet Topology			
Note	Ethernet specific attributes to the CommunicationConnector.			
Base	ARObject,CommunicationConnector,Identifiable,MultilanguageReferrable,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
networkEndpoint	NetworkEndpoint	*	ref	NetworkEndpoints

Table 3.52: EthernetCommunicationConnector

3.3.7 CDD

The System Template allows the integration of custom bus systems on the topology level.

[TPS_SYST_01127] CDD Topology support ⌈ The elements UserDefinedCluster, UserDefinedPhysicalChannel, UserDefinedCommunicationConnector and UserDefinedCommunicationController can be used to describe alternative communication technologies (e.g. I2C, USB, serial line) that are integrated in AUTOSAR as Complex Drivers. ⌋*(RS_SYST_00044)*

The Pdu-based communication via Complex Drivers is described in chapter 6.11.

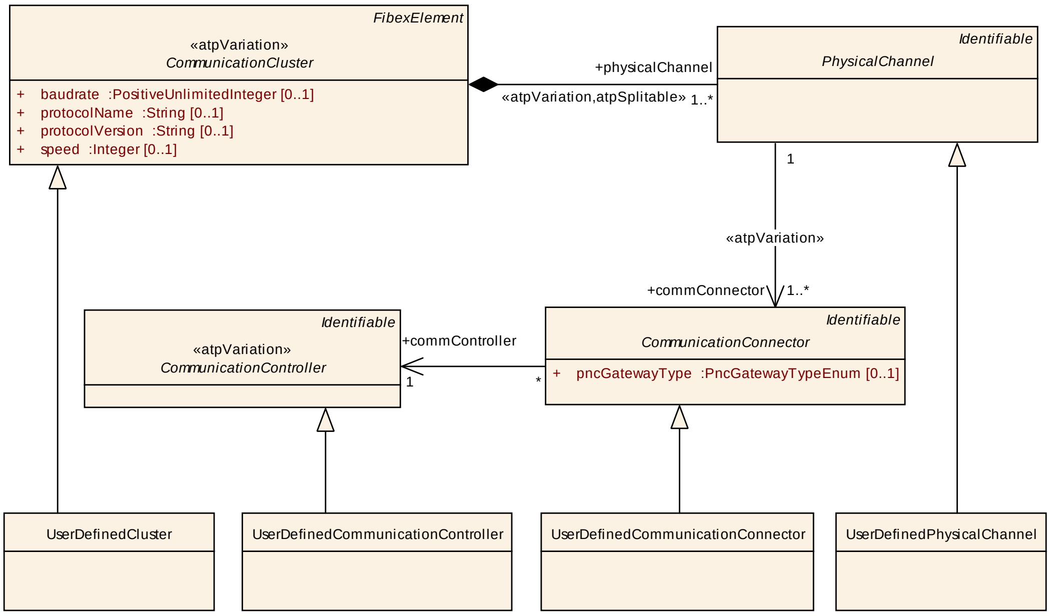

Figure 3.11: User defined topology elements

Class	≪**atpVariation**≫ **UserDefinedCluster**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::CddSupport			
Note	This element allows the modelling of arbitrary Communication Clusters (e.g. bus systems that are not supported by AUTOSAR). **Tags:** atp.recommendedPackage=CommunicationClusters			
Base	ARObject,CollectableElement,CommunicationCluster,Fibex Element,Identifiable,MultilanguageReferrable,PackageableElement,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 3.53: UserDefinedCluster

Class	UserDefinedPhysicalChannel			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::CddSupport			
Note	This element allows the modelling of arbitrary Physical Channels.			
Base	ARObject,Identifiable,MultilanguageReferrable,PhysicalChannel,Referrable			
Attribute	Datatype	Mul.	Kind	Note
–	–	–	–	–

Table 3.54: UserDefinedPhysicalChannel

Class	UserDefinedCommunicationConnector			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::CddSupport			
Note	This element allows the modelling of arbitrary Communication Connectors.			
Base	ARObject,CommunicationConnector,Identifiable,MultilanguageReferrable,Referrable			
Attribute	Datatype	Mul.	Kind	Note
–	–	–	–	–

Table 3.55: UserDefinedCommunicationConnector

Class	≪atpVariation≫ UserDefinedCommunicationController			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::CddSupport			
Note	This element allows the modelling of arbitrary Communication Controllers.			
Base	ARObject,CommunicationController,Identifiable,MultilanguageReferrable,Referrable			
Attribute	Datatype	Mul.	Kind	Note
–	–	–	–	–

Table 3.56: UserDefinedCommunicationController

3.4 Mapping of Topology Entities onto Hardware Elements

As explained in the previous sections, the System Template contains all classes necessary to describe the physical topology in an AUTOSAR system. Based on this description, the communication matrix can be realized as explained in chapter 6.

[TPS_SYST_01019] Mapping of topology elements to elements of the ECU Resource Template ⌈ It is possible to map the hardware related topology elements onto their counterpart definitions in the ECU Resource Template (Figure 3.12). ⌋*(RS_SYST_00006)*

It can be specified which `HwElement` is realizing each given `EcuInstance`, providing the means for algorithms to map software components onto the systems `EcuInstance`. By specifying which `hwCommunicationPort`[1] on a `hwCommunicationController`[2] implements the topology's `CommunicationConnector` on a `Com-`

[1] `HwPinGroup` which is of category Communication Port
[2] `HwElement` which is of category Communication Controller

municationController, the hardware-oriented parameters in the Communication-drivers may be derived in ECU configuration phase.

Please note that this is a rather specific type of mapping, optionally binding ECU-local topology elements to specific hardware resources. It should not be confused with the System Mapping part of the System Description, where system-wide mapping decisions are described, like e.g. the the mapping of Software Components onto ECUs or the mapping of Data Element Prototypes onto System Signals (for the System Mapping, see chapter 5).

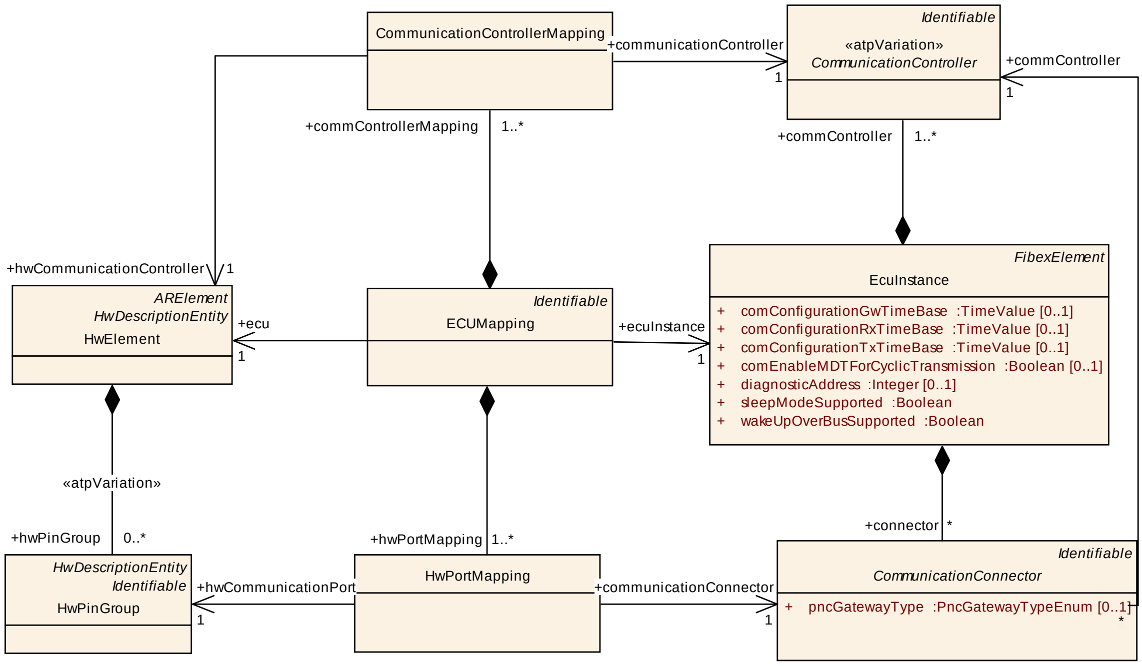

Figure 3.12: Mapping of topology description elements in the System Template onto hardware elements defined in the ECU Resource Template (ECUResourceMapping)

[constr_3006] valid EcuMapping ⌈The referenced hwCommunicationController and hwCommunicationPort shall be part of the referenced ecu.

ECUMapping.ecu.nestedElement contains ECUMapping.communicationControllerMapping.hwCommunicationController

ECUMapping.ecu.nestedElement contains ECUMapping.hwPortMapping.hwCommunicationPort ⌋

3.4.1 ECU Mapping

ECUMapping allows to assign a HwElement to an EcuInstance used in a physical topology. A HwElement of category ECU is defined in the ECU Resource Template; it provides information about the internal hardware structure of an ECU. This information can be used by the System Generator to assign or validate the mapping of Software Component Prototypes onto EcuInstances.

[TPS_SYST_01013] EcuInstance stands for its own ⌈ An EcuInstance can be defined in a stand alone and reusable way without a need to have an ECUMapping. ⌋*(RS_SYST_00013)*

[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. ⌋

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	***Datatype***	***Mul.***	***Kind***	***Note***
commControllerMapping	CommunicationControllerMapping	1..*	aggr	The ECUMapping contains the mapping of all CommunicationControllers of the ECU.
ecu	HwElement	1	ref	Reference to the Ecu description 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 3.57: ECUMapping

3.4.2 Communication Controller Mapping

[TPS_SYST_01014] Semantics of CommunicationControllerMapping ⌈ CommunicationControllerMapping specifies the HwElement to realize the specified CommunicationController in a physical topology. The information may e.g. be used during ECU configuration for configuring the hardware related parameters in the communication drivers. ⌋*(RS_SYST_00013)*

Class	**CommunicationControllerMapping**			
Package	M2::AUTOSARTemplates::SystemTemplate::ECUResourceMapping			
Note	CommunicationControllerMapping specifies the CommunicationPeripheral hardware (defined in the ECU Resource Template) to realize the specified CommunicationController in a physical topology.			
Base	ARObject			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
communicationController	CommunicationController	1	ref	Reference to the CommunicationController in the System Template

Attribute	Datatype	Mul.	Kind	Note
hwCommunicationController	HwElement	1	ref	Reference to the hwCommunicationController in the ECU Resource Template.

Table 3.58: CommunicationControllerMapping

3.4.3 HW-Port Mapping

[TPS_SYST_01015] Semantics of HwPortMapping ⌈ HwPortMapping specifies the hardware to realize the specified CommunicationConnector in a physical topology. The information may e.g. be used during ECU configuration for configuring the hardware related parameters in the communication drivers. ⌋*(RS_SYST_00013)*

Class	HwPortMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::ECUResourceMapping			
Note	HWPortMapping specifies the hwCommunicationPort (defined in the ECU Resource Template) to realize the specified CommunicationConnector in a physical topology.			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
communicationConnector	CommunicationConnector	1	ref	Reference to the CommunicationConnector in the System Template
hwCommunicationPort	HwPinGroup	1	ref	Reference to the HwPinPortGroup of category CommunicationPort. The connection to the HwCommunicationController is described in the Ecu Resource Description.

Table 3.59: HwPortMapping

4 Top-level Software Composition

One of the most important inputs for the System Generator is the knowledge about the Application Software Components, their communication capabilities and the connections between them: Each SystemSignal (chapter 6.2) that is going to be exchanged between mapped Software Components onto different ECUs is a consequence of a connection between such application Software Components.

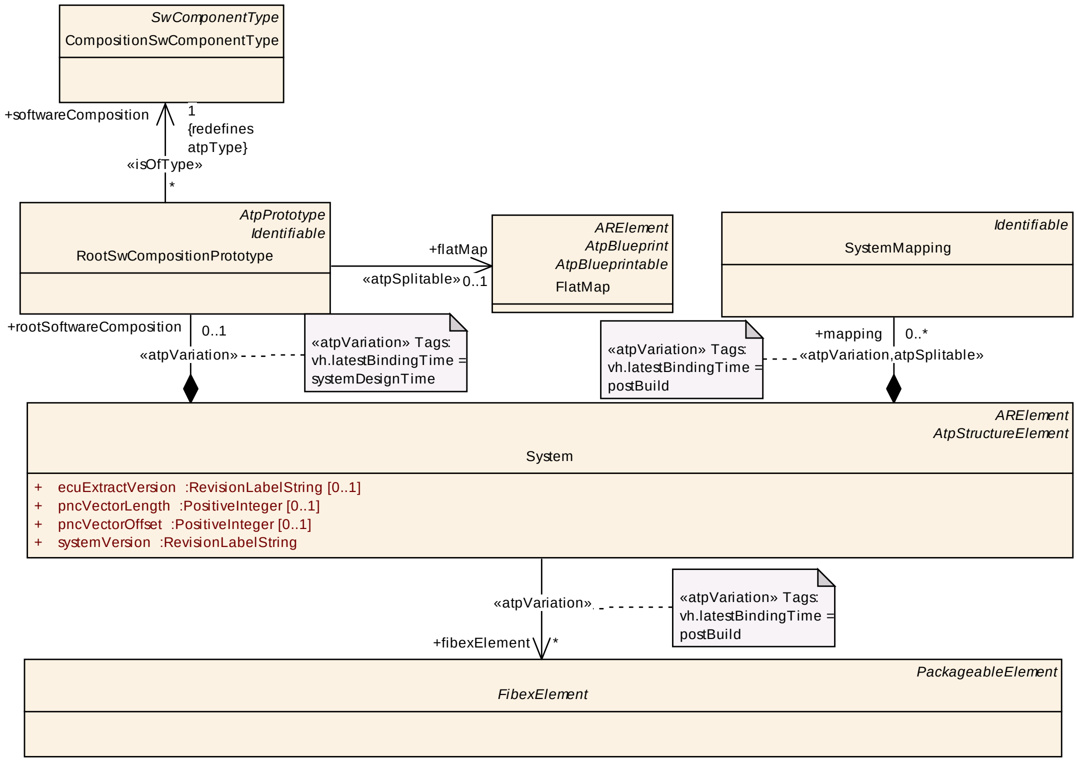

Figure 4.1: Inclusion of a (top-level) Software Composition into an AUTOSAR system (SystemTemplate)

In AUTOSAR, Software Components can either be atomic (AtomicSwComponentType) or may consist of a composition of other Software Components CompositionSwComponentType [5]. In order to assemble non-trivial applications from AUTOSAR components, such compositions can be built up hierarchically, until the outermost CompositionSwComponentType forms a kind of top-level composition.

[constr_3031] Complete System Description does not have ports ⌈ 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 SwComponentTypes, PortPrototypes, PortInterfaces, VariableDataPrototypes, InternalBehavior etc. ⌋

[TPS_SYST_01016] System Extract, Ecu System Description and Ecu Extract may have ports ⌈ In a System with category SYSTEM_EXTRACT and a

System with category ECU_SYSTEM_DESCRIPTION and a System with category ECU_EXTRACT outside ports for the outermost composition are allowed. ⌋(*RS_SYST_00027*)

Since the System/Ecu Extract represents the view on one Ecu, there may be the need to define the communication of this extract with the outside world.

Two approaches are available how the external communication of an ECU in the System Extract is described. In section 9.2 the communication mapping is performed in the hierarchical structure of software components. In section 9.3 external communication delegation ports are added to the System extract outermost composition. Each delegated port is connected via a DelegationSwConnector with ports of the included components that are used for the external communication.

A System considers such a top-level CompositionSwComponentType as its application software system input by owning exactly one RootSwCompositionPrototype class, which points to the CompositionSwComponentType forming the input via its ≪isOfType≫ relationship as shown in Figure 4.1.

[TPS_SYST_01017] The role of the top-level software composition ⌈ An AUTOSAR System uses the specialized prototype class RootSwCompositionPrototype in order to designate the referenced CompositionSwComponentType as the top-level software composition. ⌋(*RS_SYST_00006*)

Class	**RootSwCompositionPrototype**			
Package	M2::AUTOSARTemplates::SystemTemplate			
Note	The RootSwCompositionPrototype represents the top-level-composition of software components within a given System. According to the use case of the System, this may for example be the a more or less complete VFB description, the software of a System Extract or the software of a flat ECU Extract with only atomic SWCs. Therefore the RootSwComposition will only occasionally contain all atomic software components that are used in a complete VFB System. The OEM is primarily interested in the required functionality and the interfaces defining the integration of the Software Component into the System. The internal structure of such a component contains often substantial intellectual property of a supplier. Therefore a top-level software composition will often contain empty compositions which represent subsystems. The contained SwComponentPrototypes are fully specified by their SwComponentTypes (including PortPrototypes, PortInterfaces, VariableDataPrototypes, SwcInternalBehavior etc.), and their ports are interconnected using SwConnectorPrototypes.			
Base	ARObject,AtpFeature,AtpPrototype,Identifiable,MultilanguageReferrable,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
calibrationParameterValueSet	CalibrationParameterValueSet	*	ref	Used CalibrationParameterValueSet for instance specific initialization of calibration parameters. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=calibrationParameterValueSet

Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
flatMap	FlatMap	0..1	ref	The FlatMap used in the scope of this RootSwCompositionPrototype. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=flatMap
softwareComposition	CompositionSwComponentType	1	tref	We assume that there is exactly one top-level composition that includes all Component instances of the system **Stereotypes:** isOfType

Table 4.1: RootSwCompositionPrototype

5 Mapping

A central part of the system generation process is the mapping of software components (SwComponentPrototypes) to ECUs, and the subsequent mapping of the communication between these software components to bus frames. Input to the software component mapping is the RootSwCompositionPrototype, which describes which software components have to be mapped, and the System Topology, which defines the ECU instances that are available as mapping targets. Once this mapping is done, also the communication matrix has to be taken into account for the next mapping step, the mapping of data elements exchanged between software components to bus frames. This communication matrix may either be predefined, or may be generated as part of this second mapping step. In the metamodel, different aspects of these mapping are aggregated by the meta class SystemMapping, as shown in Figure 5.1.

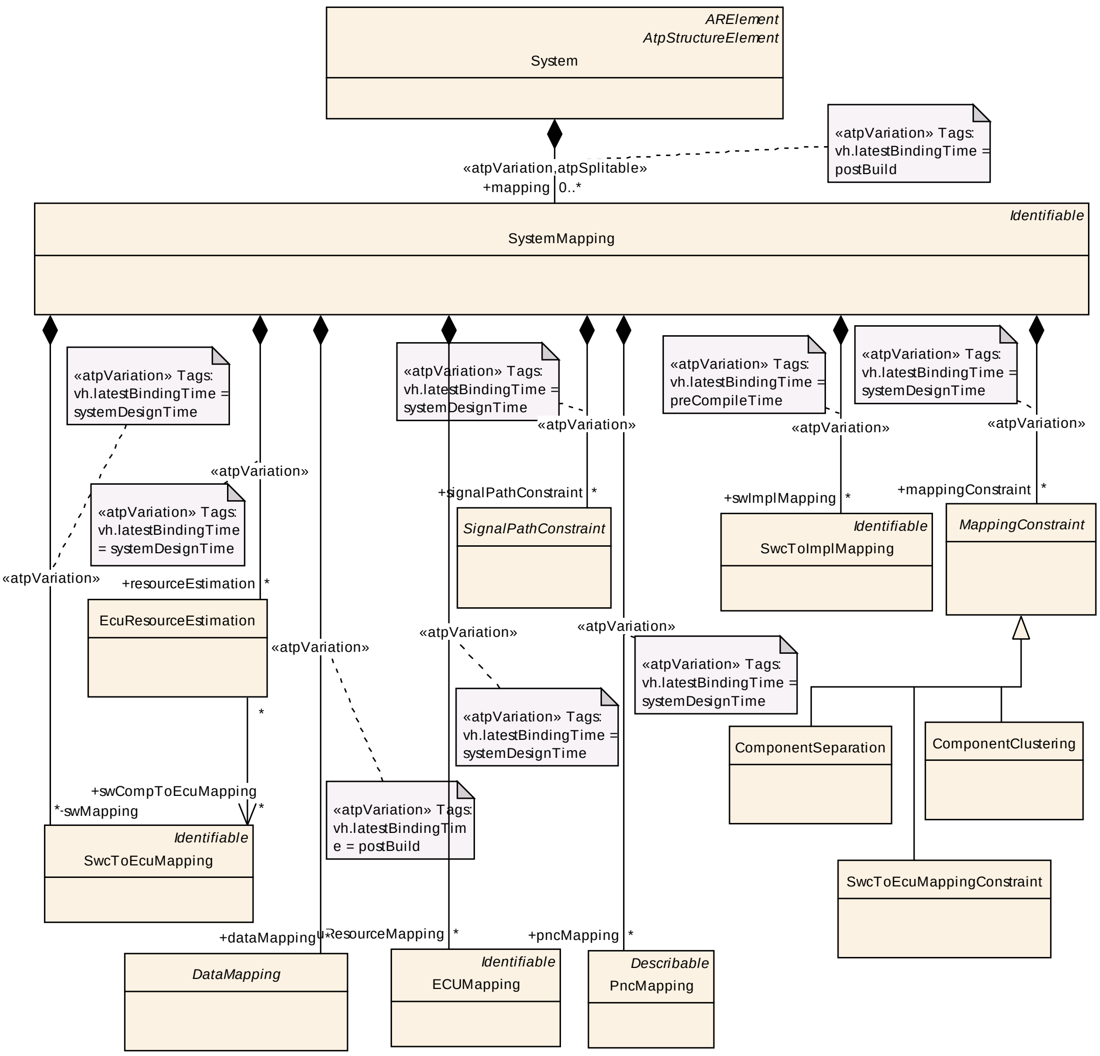

Figure 5.1: Mapping Overview (Mapping)

The following mappings are defined:

- The SwcToEcuMapping meta-class maps one or several SwComponentPrototypes to ECUs. In the System Constraint Description it is possible to predefine the mapping of SwComponentPrototypes to ECUs. The predefinition limits the system architect's freedom to map software components to arbitrary ECUs. After the system generation in the System Configuration Description, all atomic software components that are directly or indirectly part of the top level composition must be mapped with this mapping rule. Software component mapping is described in detail in chapter 5.1.
- The SwcToImplMapping meta-class is used to assign one Implementation to one or more SwComponentPrototypes (see chapter 5.1.2).
- The MappingConstraint meta-class is used to define constraints that constrain the mapping of software components. It's sub-classes allow to constraint which SwComponentPrototypes must be mapped together on the same ECU (ComponentClustering) and which must not be mapped to the same ECU (ComponentSeparation). The mapping constraints are described in detail in chapter 5.1.3.
- The DataMapping meta-class is used to map VariableDataPrototypes and ClientServerOperations in software component ports (i.e. the data exchanges between software components) to signals. The data mapping is described in detail in chapter 5.2.
- The SignalPathConstraint meta-class is used to define which specific way a signal (data element or client server operation arguments) between two Software Components should take in the network without defining in which frame and with which timing it is transmitted. This Signal Path Constraint is introduced in chapter 5.2.2.
- The ECUMapping meta-class is used to map the hardware related topology elements onto their counterpart definitions in the ECU Resource Template (see chapter 3.4).
- The PncMapping defines the Partial Network behavior (see chapter 5.4).
- Finally, meta-class EcuResourceEstimation specifies the resource estimation for RTE and basic software (see chapter 5.3).

Class	SystemMapping			
Package	M2::AUTOSARTemplates::SystemTemplate			
Note	The system mapping aggregates all mapping aspects (mapping of SW components to ECUs, mapping of data elements to signals, and mapping constraints).			
Base	ARObject,Identifiable,MultilanguageReferrable,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
dataMapping	DataMapping	*	aggr	The data mappings defined. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
ecuResourceMapping	ECUMapping	*	aggr	Mapping of hardware related topology elements onto their counterpart definitions in the ECU Resource Template. atpVariation: The ECU Resource type might be variable. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=systemDesignTime
mappingConstraint	MappingConstraint	*	aggr	Constraints that limit the mapping freedom for the mapping of SW components to ECUs. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=systemDesignTime
pncMapping	PncMapping	*	aggr	**Stereotypes:** atpVariation**Tags:** vh.latestBindingTime=systemDesignTime
resourceEstimation	EcuResourceEstimation	*	aggr	Resource estimations for this set of mappings, zero or one per ECU instance. atpVariation: Used ECUs are variable. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=systemDesignTime
signalPathConstraint	SignalPathConstraint	*	aggr	Constraints that limit the mapping freedom for the mapping of data elements to signals. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=systemDesignTime
swImplMapping	SwcToImplMapping	*	aggr	The mappings of AtomicSoftwareComponentInstances to Implementations. atpVariation: Derived, because SwcToEcuMapping is variable. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
swMapping	SwcToEcuMapping	*	aggr	The mappings of SW components to ECUs. atpVariation: SWC shall be mapped to other ECUs. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=systemDesignTime

Table 5.1: SystemMapping

5.1 Software Component Mapping

A fundamental concept of AUTOSAR is that SW components may be developed independently of a specific ECU hardware, and can be mapped to an ECU in the AUTOSAR System Generation Process. The System Constraint Description acts as an input to this System Generation Phase. Nevertheless, there may be some SW components which are already mapped due to previous iterations of the system generation step, and there may be system constraints that limit the system architect's freedom to map SW components to arbitrary ECUs. In the following, the individual elements are described in more detail.

5.1.1 SW Component to ECU Mapping

[TPS_SYST_01001] Definition of `SwcToEcuMapping` ⌈ With the `SwcToEcuMapping` element it is possible to express the mapping of `SwComponentPrototype`s to one `EcuInstance` or optional to individual `HwElement`s with `category` `Processing Unit` residing in this ECU. An optional assignment to defined `EcuPartition`s (memory partitions) is also possible, as well as the assignment of Sensor/Actuator `SwComponentPrototype`s to Sensor/Actuator `HwElement`s. ⌋*(RS_SYST_00007, RS_SYST_00033)*

The mapping to cores and memory partitions enables to express the architectural requirements/constraints, especially related to safety. For example, it may be required that some SWCs shall run on different partitions or cores in the same ECU, or shall run on different ECUs. Figure 5.2 shows this structure. The predefinition will force the system generator to use the specified mapping.

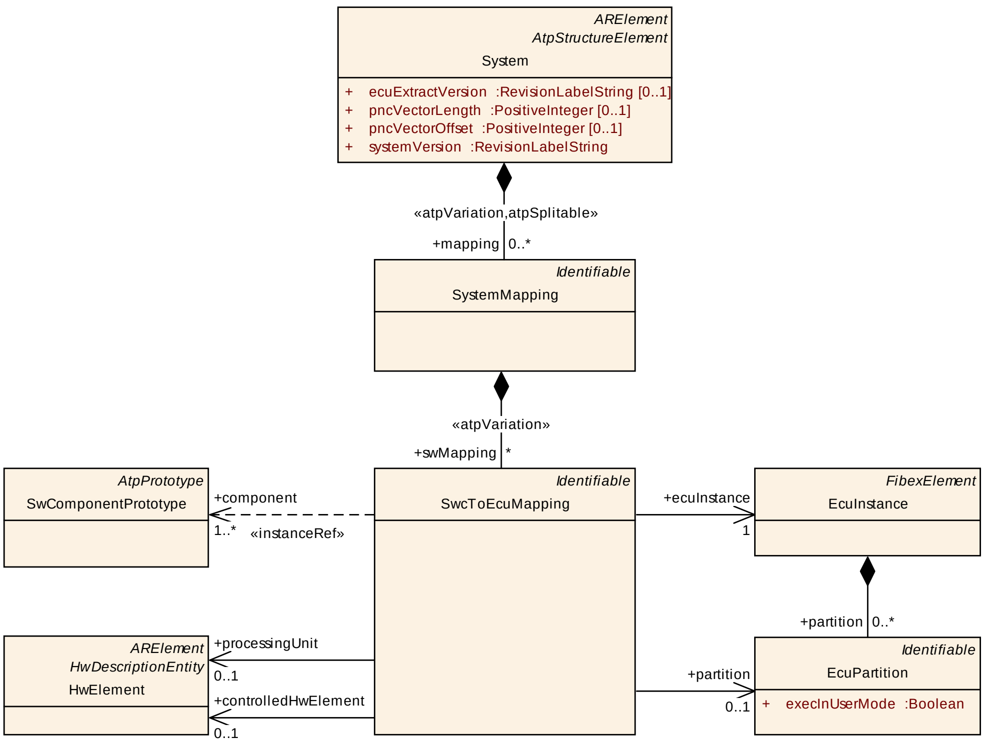

Figure 5.2: SW component to ECU mapping (SwcToEcuMapping)

The SwcToEcuMapping collects a list of all SwComponentPrototypes that shall be deployed onto the associated SwcToEcuMapping targets.

[constr_3032] Combinations of SwcToEcuMapping targets ⌈ For each combination of EcuInstance and the optional processingUnit and the optional partition and the optional controlledHwElement one SwcToEcuMapping shall be used. ⌋

SwcToEcuMapping may map either prototypes of AtomicSwComponentType or those of CompositionSwComponentType.

[TPS_SYST_01020] Unconditional mapping of atomic Software Components ⌈ In case a prototype of an atomic Software Components is mapped, the mapping is unconditional. ⌋*(RS_SYST_00007)*

[TPS_SYST_01021] Mapping of CompositionSwComponentType ⌈ In case a mapped SwComponentPrototype refers to a CompositionSwComponentType, the mapping is applied to any inner SwComponentPrototype recursively; however, it may be overwritten by additional SwcToEcuMapping mapping inner SwComponentPrototype to different EcuInstances. ⌋*(RS_SYST_00007)*

Usually a particular component prototype can be mapped explicitly to at most one ECU in a given system (leaving aside variant handling and the implicit mapping of "inner" prototypes mentioned above) but there are two exceptions:

- **[TPS_SYST_01022] Prototype of a ParameterSwComponentType can be mapped to more than one ECU** ⌈ A prototype of a ParameterSwComponentType can be mapped to more than one ECU. This is required, because this special component does not communicate over the network, so that a copy of the prototype has to be created on each ECU were it is required. ⌋*(RS_SYST_00007)*
- **[TPS_SYST_01023] Prototype of an ServiceProxySwComponentType can be mapped to more than one ECU** ⌈ A prototype of an ServiceProxySwComponentType can be mapped to more than one ECU even if it appears only once in the VFB system, because a prototype of this special component is required on each ECU, for which local Services are addressed via the proxy. ⌋*(RS_SYST_00031)*

[constr_3021] Mapping of SensorActuatorSwComponents to SensorActuator HwElements ⌈Only SwComponentPrototypes that are typed by SensorActuatorSwComponentType shall be mapped to a HwElement with category SensorActuator via the controlledHwElement relation. ⌋

The following table describes the SwcToEcuMapping in detail.

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 EcuPartition. For each combination of ECUInstance and the optional ProcessingUnit and the optional EcuPartition and the optional SensorActuator only one SwcToEcuMapping shall be used.			
Base	ARObject,Identifiable,MultilanguageReferrable,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
component	SwComponentPrototype	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 SwComponentPrototype INSIDE the Composition to another ECU Instance the inner mapping overrides the outer mapping.
controlledHwElement	HwElement	0..1	ref	Optional mapping of SwComponentPrototypes that are typed by SensorActuatorSwComponentType to a HwElement with category SensorActuator.
ecuInstance	EcuInstance	1	ref	EcuInstance is a reference to an ECU Instance description
partition	EcuPartition	0..1	ref	An optional mapping of SWCs to Partitions. With this mapping an OEM has the option to predefine an allocation in the System Design phase. The final and complete assignment is described in the OS Configuration.
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 ProcessingUnit.

Table 5.2: SwcToEcuMapping

5.1.2 Software Component to Implementation Mapping

As several implementations may exist for the same AtomicSwComponentType, it needs to be decided on and specified which instances of a given AtomicSwComponentType are mapped to which Implementation. According to the AUTOSAR Methodology this information can either be added within the Configure System activity, or later when the RTE part is configured during Configure ECU phase. If the mapping is done in System Configuration, a SwcToImplMapping is being used for assigning one Implementation to one or more instances of SwComponentPrototype relating to the same AtomicSwComponentType. This is illustrated in Figure 5.3.

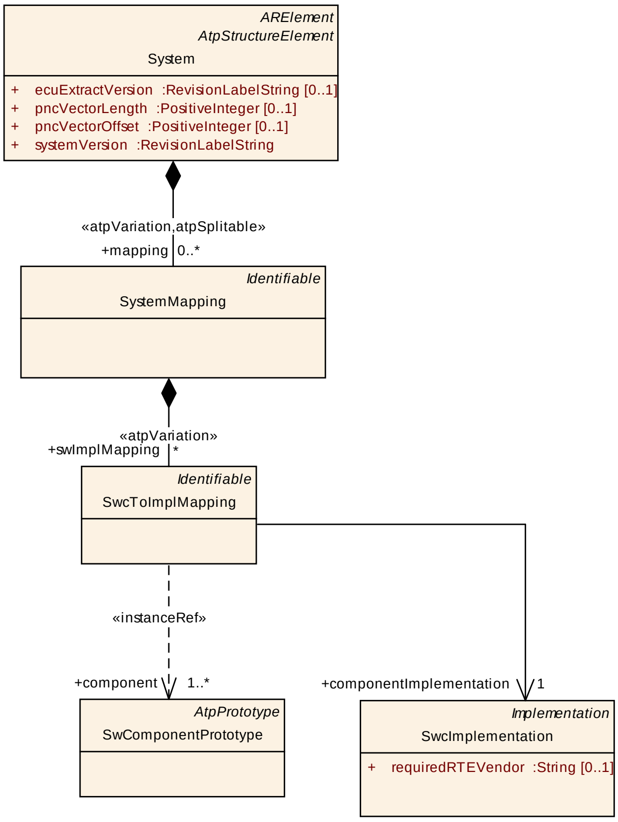

Figure 5.3: SW Component to Implementation mapping (SwcToImplMapping)

[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 ⌋

The following table contains the detailed description of `SwcToImplMapping`:

Class	**SwcToImplMapping**			
Package	M2::AUTOSARTemplates::SystemTemplate::SWmapping			
Note	Map instances of an AtomicSwComponentType to a specific Implementation.			
Base	ARObject,Identifiable,MultilanguageReferrable,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
component	SwComponentPrototype	1..*	iref	Reference to the software component instances that are being mapped to the specified Implementation. The targeted SwComponentPrototype needs be of the AtomicSwComponentType being implemented by the referenced Implementation.
componentImplementation	SwcImplementation	1	ref	Reference to a specific Implementation description. Implementation to be used by the specified SW component instance. This allows to achieve more precise estimates for the resource consumption that results from mapping the instance of an atomic SW component onto an ECU.

Table 5.3: SwcToImplMapping

5.1.3 Software Component Mapping Constraints

In contrast to the mapping description described in the previous chapters, mapping constraints allow to define invariants that have to be fulfilled by a valid mapping. They are aggregated in the MappingConstraint element as introduced in chapter 5 and depicted Figure 5.1. This chapter describes which mapping constraints can be described in the System Constraint Description. The description of this meta-class can be found in the following table:

Class	**MappingConstraint (abstract)**			
Package	M2::AUTOSARTemplates::SystemTemplate::SWmapping			
Note	Different constraints that may be used to limit the mapping of SW components to ECUs.			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
introduction	DocumentationBlock	0..1	aggr	This represents introductory documentation about the mapping constraint.

Table 5.4: MappingConstraint

The two constraints (ComponentClustering and ComponentSeparation) shown in Figure 5.4 express the restrictions that Software Components impose on each other when performing the mapping onto the ECUs. In fact, before the mapping process begins, it can be useful to impose the allocation of a predefined set of SW components onto the same ECU, especially if such a set is tightly linked from a functional point of view. In the same way, two critical SW components, performing some kind of redundancy, may be not suitable to run both on the same ECU. Thus, we call these two kinds of mapping constraints, respectively, ComponentClustering and ComponentSeparation.

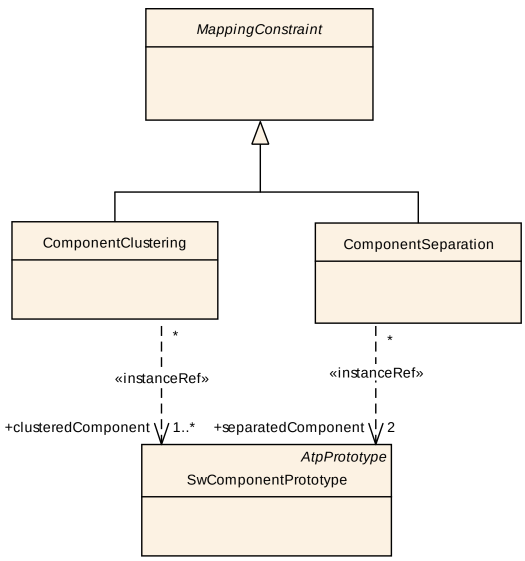

Figure 5.4: Details on ComponentClustering and ComponentSeparation (SwcClustering)

5.1.3.1 ComponentClustering

[TPS_SYST_01024] Component Clustering ⌈ The ComponentClustering constraint (also, *clustering*) is to be used for expressing that a certain set of SW components (atomic or not) must be mapped (allocated) onto the same ECU. ⌋*(RS_SYST_00008)*

This is some kind of "execute together on same ECU" constraint.

The semantic of the clustering constraint is straightforward if all referenced SW components are atomic. Otherwise, it shall be interpreted as follows:

[TPS_SYST_01025] Clustering of Compositions ⌈ All of the atomic SW components making up the composition must be mapped onto the same ECU together with all other SW components (atomic or not) referenced by the constraint. ⌋*(RS_SYST_00008)*

This also means that a *clustering* constraint can also refer to only a single composition.

A *clustering* constraint is part of a MappingConstraint element and it must refer to one or more SwComponentPrototype elements, representing the instances of the SW component(s) that must be mapped together.

Class	**ComponentClustering**			
Package	M2::AUTOSARTemplates::SystemTemplate::SWmapping			
Note	Constraint that forces the mapping of all referenced SW component instances to the same ECU			
Base	ARObject,MappingConstraint			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
clusteredComponent	SwComponentPrototype	1..*	iref	Reference to the components that have to be mapped together.

Table 5.5: ComponentClustering

5.1.3.2 ComponentSeparation

[TPS_SYST_01045] Component Seperation ⌈ The ComponentSeparation constraint (also, *separation*) is to be used for expressing that two SW components (atomic or not) shall not be mapped (allocated) onto the same ECU. ⌋*(RS_SYST_00009)*

This is some kind of “do not execute together on same ECU” constraint.

The semantic of the separation constraint is straightforward if one or both SW components are atomic. Otherwise, it shall be interpreted as follows:

[TPS_SYST_01026] Separation of Compositions ⌈ Any of the atomic SW components making up the first composition, must not be mapped onto the same ECU with any atomic SW component from the second composition. ⌋*(RS_SYST_00009)*

As a consequence, and to preserve consistency, an atomic SW component instance cannot be part of two compositions concerned by the same separation constraint, i.e. the two compositions have to be disjoint with regards to component instances[1].

A *separation* constraint is part of a MappingConstraint element and it must refer to two SwComponentPrototype elements, representing the two SW component instances that must not be allocated together.

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. 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	**Datatype**	**Mul.**	**Kind**	**Note**
separatedComponent	SwComponentPrototype	2	iref	The two components that have to be mapped to different ECUs

Table 5.6: ComponentSeparation

[constr_3004] Clustering and separation must be exclusive ⌈Clustering and separation must be exclusive, i.e. it SHALL NOT be possible that two SwComponentPrototypes A and B are associated by a ComponentClustering and by a ComponentSeparation. ⌋

[1] The only case where a component instance could be in both sets is if the ComponentSeparation refers to two elements where one of them is a substructure of the other. Consider the case that Atomic SW Component A is aggregated by composition B, which in turn is aggregated by composition C. Then instance A is both in B and C. It is not a good idea to formulate a separation constraint stating that B and C should not be on the same ECU.

5.1.3.3 SwcToEcuMappingConstraint

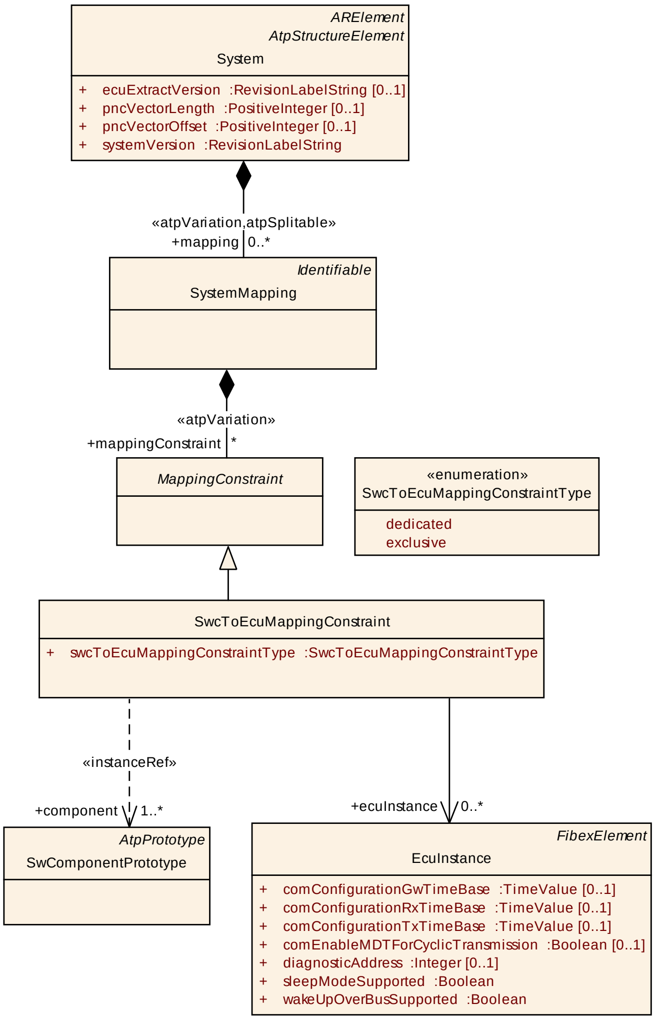

Figure 5.5: Dedicated and exclusive Mapping of SWC to ECUs

The SwcToEcuMappingConstraint shown in Figure 5.5 allows to restrict the mapping of SW components to ECUs.

[TPS_SYST_01027] Mapping of specific SW components to dedicated Ecus ⌈ If the swcToEcuMappingConstraintType is set to dedicated, the constraint expresses that the mapping of specific SW components is only allowed to one of a number of dedicated ECUs. The mapping to other ECUs is not allowed. ⌋*(RS_SYST_00011)*

[TPS_SYST_01028] Task of the System Generator ⌈ When the system generator performs the mapping of software components to ECUs it has to take the MappingConstraints into account. ⌋

[TPS_SYST_01029] Mapping of specific SW components to exclusive Ecus ⌈ If the swcToEcuMappingConstraintType is set to exclusive, it means that the referenced software components cannot be mapped to the referenced ECUs. ⌋*(RS_SYST_00010)*

With these kinds of constraints, no fixed mapping of a software component to an ECU is performed. Instead, they can be seen as invariants that have to be fulfilled when the actual SWC mapping using `SwcToEcuMapping` is performed.

Class	**SwcToEcuMappingConstraint**			
Package	M2::AUTOSARTemplates::SystemTemplate::SWmapping			
Note	The System Constraint Description has to describe dedicated and exclusive mapping of SW-Cs to one or more ECUs. Dedicated mapping means that the SW-C can only be mapped to the ECUs it is dedicated to. Exclusive Mapping means that the SW-C cannot be mapped to the ECUs it is excluded from.			
Base	ARObject,MappingConstraint			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
component	SwComponentPrototype	1	iref	Reference to SwComponentPrototypes for which the dedicated or exclusive mapping is defined.
ecuInstance	EcuInstance	*	ref	If the dedicated mapping is described, the SwComponentPrototypes can only be mapped to these referenced ECUInstances. If the exclusive mapping is described, the SwComponentPrototypes cannot be mapped to these referenced ECUInstances.
swcToEcuMappingConstraintType	SwcToEcuMappingConstraintType	1	attr	This attribute determines if dedicated or exclusive mapping is used.

Table 5.7: SwcToEcuMappingConstraint

Enumeration	**SwcToEcuMappingConstraintType**
Package	M2::AUTOSARTemplates::SystemTemplate::SWmapping
Note	There are two different SwcToEcuMapping constraints: dedicated mapping and exclusive mapping.
Literal	*Description*
dedicated	Dedicated mapping means that the SW-C can only be mapped to the ECUs it is dedicated to.
exclusive	Exclusive mapping means that the SW-C cannot be mapped to the ECUs it is excluded from.

Table 5.8: SwcToEcuMappingConstraintType

5.2 Data Mapping

The data mapping description may either be mapping of client server communication or sender receiver communication (see Figure 5.6). It is used to map VariableDataPrototypes or ClientServerOperations of SW Component Ports to SystemSignals.

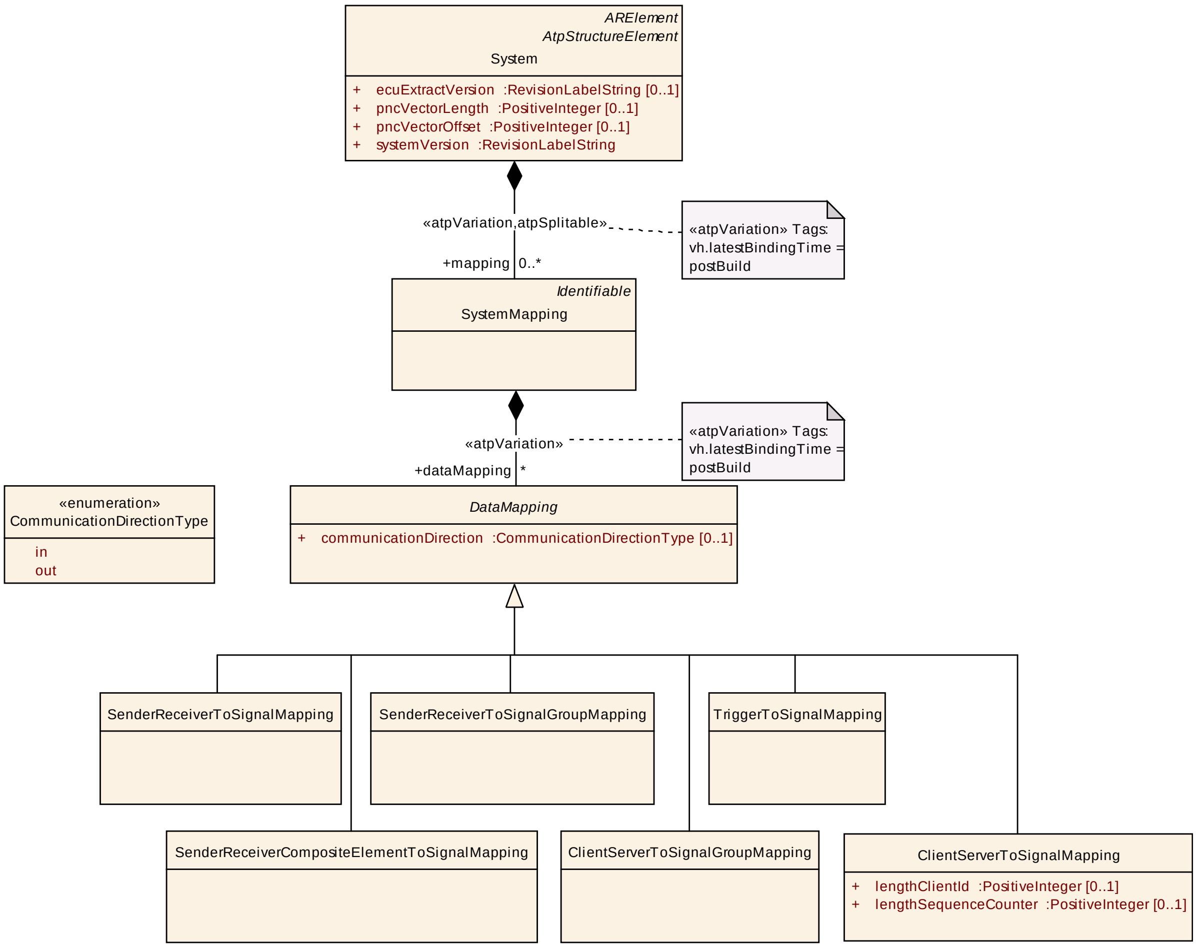

Figure 5.6: Overview: Data Mapping Description (DataMappingOverview)

[TPS_SYST_01030] Representation of VariableDataPrototypes and ClientServerOperations in System Description ⌈ SystemSignals represent VariableDataPrototypes and ClientServerOperations in the communication description. ⌋*(RS_SYST_00025)*

[TPS_SYST_01032] Independence of SystemSignals from CommunicationClusters ⌈ The SystemSignals can be defined independently of CommunicationClusters. ⌋

This chapter describes how the VariableDataPrototypes and ClientServerOperations are mapped onto SystemSignals. The Communication chapter (6) describes how the SystemSignals are mapped into Pdus and Frames, implementing the actual inter-ECU communication.

Class	**DataMapping (abstract)**			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	Mapping of port elements (data elements and parameters) to frames and signals.			
Base	ARObject			
Attribute	***Datatype***	***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 DataMapping.
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 5.9: DataMapping

[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 in chapter 6.7.5. ⌋

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,PackageableElement,Referrable			
Attribute	***Datatype***	***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 5.10: SystemSignal

A SystemSignal is used to represent VariableDataPrototypes for network transport.

[TPS_SYST_01144] Physical properties of a SystemSignal ⌈ With the aggregation of SwDataDefProps in the role physicalProps the physical properties of the SystemSignal can be specified. ⌋

[TPS_SYST_05000] System Description doesn't use a complete Software Component Description ⌈ If the System Description doesn't use a complete Software Component Description (VFB View) the data mapping of VariableDataPrototypes or ArgumentDataPrototypes owned by ClientServerOperations on SystemSignals does not need to be defined. This supports the inclusion of legacy signals. ⌋*(RS_SYST_00001)*

[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_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_3049] Role of SystemSignal in inter-ECU client server communication with clients located on different ECUs ⌈In case of a n:1 inter-ECU client server communication with clients located on different ECUs different SystemSignals shall be used for each Ecu. ⌋

[TPS_SYST_01087] Role of SystemSignal in inter-ECU client server communication with clients located on the same ECU ⌈ In case of n:1 inter-ECU client server communication it is allowed to use the same SystemSignal for several clients on the same Ecu, if the client identifier is used to distinguish the different clients. ⌋

[TPS_SYST_02011] initValues of receivers that are mapped to the same Ecu ⌈ All receivers of a given SystemSignal on the same EcuInstance shall have identical initValues. ⌋

[constr_3074] No TransmissionAcknowledgementRequest for multiple senders ⌈ If more than one SenderComSpec exist (in different PortPrototypes on atomic level) that refer to data elements effectively mapped to the same SystemSignal it is not allowed that any SenderComSpec aggregates transmissionAcknowledge. ⌋

Please note that the term "effectively mapped" refers to the fact that the DataMapping can refer to a dataElement in a "delegation" PortPrototype on the surface of a rootSwComposition of an Ecu Extract OR to PortPrototypes inside the rootSwComposition. Both ways shall be considered.

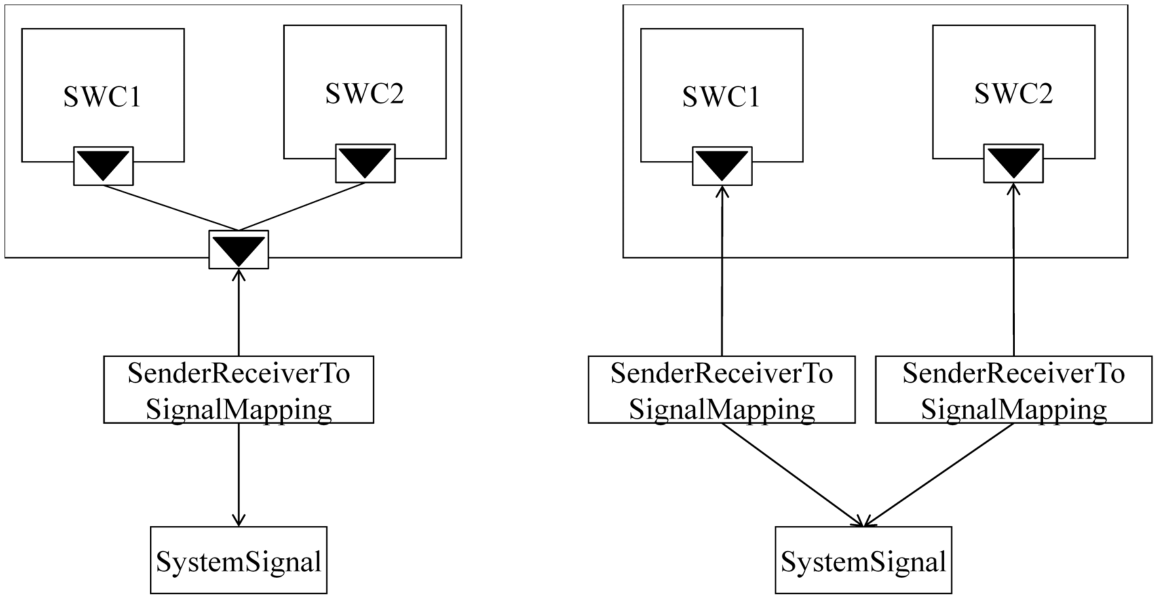

Figure 5.7: Example for data elements that are effectively mapped to the same SystemSignal

The different kinds of data mapping are described in the following sections in detail.

5.2.1 Mapping of Variable Data Prototypes on System Signals

This section describes how VariableDataPrototypes are mapped onto SystemSignals. For a detailed description of the interconnection of software components refer to [5].

It is the task of system configuration to map VariableDataPrototype, ClientServerOperation, or Trigger contained in PortPrototypes referenced by the SwConnector onto a SystemSignal.

[TPS_SYST_01033] DataMapping and SwConnector ⌈ For the purpose of creating DataMappings PortPrototypes may or may not be connected by SwConnectors.
⌋

The same SystemSignal may satisfy more than one SwConnector (1:n communication), and one SwConnector may be implemented by several SystemSignals (e.g. one per VariableDataPrototype in the PortInterface being connected), so there is no 1:1 mapping between SwConnectors and SystemSignals.

In the following sections, each reference to a VariableDataPrototype, ArgumentDataPrototype, or Trigger is of type AtpInstanceRef [2]. This means it not only references the actual VariableDataPrototype, but additionally contains contextual references to the PortPrototype and the hierarchy of SwComponentPrototypes forming the individual instance context of the VariableDataPrototype.

The following rules are valid for the mapping of VariableDataPrototypes, ClientServerOperations, or Triggers on SystemSignals:

[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 ⌋

Please note that for DataMappings the following use cases are supported:

- Sending: one DataMapping that point to the DataPrototype and to the SystemSignal with communicationDirection out.
- Receiving: one DataMapping that point to the DataPrototype and to the SystemSignal with communicationDirection in.
- Sending and Receiving: two DataMappings that point to the DataPrototype and to the SystemSignal, one with communicationDirection in and one with communicationDirection out.

[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 ⌋

In a complete System with category SYSTEM_DESCRIPTION, it is sufficient to refer to the VariableDataPrototype in the PPortPrototype or the RPortPrototype to define the mapping of the communication between a provider and its receivers.

This is possible since the connectors implicitly define which RPortPrototype are connected to which PPortPrototype. In case the System with category SYSTEM_DESCRIPTION does not use a complete Software Component Description (VFB View) the data mapping needs not to be defined. This supports the inclusion of legacy signals.

[TPS_SYST_01137] Several DataMappings may be defined for the same SystemSignal ⌈ For a SystemSignal which is

- part in several SystemSignalGroups
- part in at least one SystemSignalGroup and at the same time is transmitted additionally as standalone SystemSignal

several DataMappings may be defined. ⌋

[TPS_SYST_01050] SystemSignal in the System Extract and ECU Extract ⌈ In the System with category SYSTEM_EXTRACT or ECU_EXTRACT the missing DataMappings on the complementary Sender/Receiver side needs to be supplemented. ⌋

In the System with category SYSTEM_EXTRACT or ECU_EXTRACT, where only the relevant parts of the rootSoftwareComposition are defined, it is necessary to utilize the information from the complementary PortPrototype if the corresponding PortPrototype is located on another ECU and thus is not part of the extract. This is described in more detail in chapter 9.2 and chapter 10.2.3.

Therefore in a System with category ECU_EXTRACT the DataMappings are provided on both, PPortPrototypes and RPortPrototypes.

[TPS_SYST_01034] Data Mappings can be applied to compositions and atomic software components ⌈ DataMappings can be applied to CompositionSwComponentTypes and on AtomicSwComponentTypes. ⌋

[TPS_SYST_01035] Transformation of Data Mappings during flattening ⌈ During the creation of the System with category ECU_EXTRACT (flattening) the existing DataMappings that refer to CompositionSwComponentTypes shall be transformed to refer to AtomicSwComponentTypes instead. ⌋

[TPS_SYST_01036] No additional Data Mappings in composition substructure ⌈ When a CompositionSwComponentType is refined by a supplier the already existing DataMappings that refer to the CompositionSwComponentType shall not be copied to the internal substructure. ⌋

Suppliers who add substructure to a CompositionSwComponentType by adding SwComponentPrototypes and SwConnectors shall respect the predefined DataMappings on the CompositionSwComponentType.

The OEM/Supplier Collaboration Scenario is described in chapter 9.1.

[constr_3087] DataMapping to PRPortPrototype ⌈ For inter-ECU communication between SwComponentPrototypes which involves PRPortPrototypes for each DataPrototype there shall be one SystemSignal and at most two DataMappings, one for each direction. ⌋

Please note that for DataMappings the following use cases are supported:

- Sending: one DataMapping that point to the DataPrototype and to the SystemSignal with communicationDirection out.

- Receiving: one DataMapping that point to the DataPrototype and to the SystemSignal with communicationDirection in.
- Sending and Receiving: two DataMappings that point to the DataPrototype and to the SystemSignal, one with communicationDirection in and one with communicationDirection out.

[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.

⌋

5.2.1.1 Mapping of Variable Data Prototypes with primitive datatypes on System Signals (Sender-Receiver Communication)

This section describes the relation between the VariableDataPrototype with primitive datatypes and the SystemSignal (see Figure 5.8).

[constr_3505] Criteria for primitive Data Mapping ⌈ The VariableDataPrototype referenced by dataElement shall be typed by one of

- ApplicationPrimitiveDataType of category VALUE, BOOLEAN, and STRING and for which a DataTypeMappingSet exists that points to an ImplementationDataType that fulfills all of the following conditions:
  - The ImplementationDataType is either
    - ∗ of category TYPE_REFERENCE that eventually references an ImplementationDataType of category VALUE **or**
    - ∗ the ImplementationDataType is of category VALUE.
  - The ImplementationDataType either
    - ∗ represents the platform type uint8 **or**

    * references a SwBaseType with a SwBaseType.baseTypeDefinition.baseTypeSize set to value 8 and the SwBaseType.baseTypeDefinition.baseTypeEncoding set to NONE.

- ImplementationDataType of category ARRAY that has a subElement that fulfills all of the following conditions:
  - the subElement is either
    * of category TYPE_REFERENCE that (by reference to a swDataDefProps.implementationDataType) eventually references an ImplementationDataType of category VALUE **or**
    * the subElement is of category VALUE.
  - the subElement (by reference to a swDataDefProps.implementationDataType) either
    * implements the platform type uint8 **or**
    * references a SwBaseType with a SwBaseType.baseTypeDefinition.baseTypeSize set to the value 8 and the SwBaseType.baseTypeDefinition.baseTypeEncoding set to NONE.
- ApplicationArrayDataType for which a DataTypeMap exists that points to an ImplementationDataType that fulfills the above mentioned condition.

Alternatively, the following rules apply for a scenario where a DataTypeMap does not yet exist:

The VariableDataPrototype referenced by argument shall be typed by one of

- ApplicationPrimitiveDataType of category BOOLEAN
- ApplicationPrimitiveDataType of category VALUE if the following conditions are fulfilled:
  - ApplicationPrimitiveDataType.swDataDefProps.dataConstr exists and refers to a PhysConstrs.
  - ApplicationPrimitiveDataType.swDataDefProps.compuMethod exists and refers to a CompuMethod of category TEXTTABLE and CompuMethod.compuPhysToInternal exists.
  - Application of ApplicationPrimitiveDataType.swDataDefProps.compuMethod to ApplicationPrimitiveDataType.swDataDefProps.dataConstr yields a numerical range in [0 .. 255].
- ApplicationPrimitiveDataType of category STRING if
  - ApplicationPrimitiveDataType.swDataDefProps.swRecordLayout exists and values of SwRecordLayout.swRecordLayoutGroup.swRecordLayoutGroupFrom and SwRecordLayout.swRecordLayoutGroup.swRecordLayoutGroupTo are both set to 1.

  - ApplicationPrimitiveDataType.swDataDefProps.swTextProps exists and refers to an SwBaseType where the SwBaseType.baseTypeDefinition.baseTypeEncoding is set to NONE and the value of SwBaseType.baseTypeDefinition.baseTypeSize is set to 8.

- ApplicationArrayDataType where the aggregated element fulfills the following conditions:
  - ApplicationPrimitiveDataType.swDataDefProps.dataConstr exists and refers to a PhysConstrs.
  - ApplicationPrimitiveDataType.swDataDefProps.compuMethod exists and refers to a CompuMethod of category TEXTTABLE and CompuMethod.compuPhysToInternal exists.
  - Application of ApplicationPrimitiveDataType.swDataDefProps.compuMethod to ApplicationPrimitiveDataType.swDataDefProps.dataConstr yields a numerical range in [0 .. 255].

⌋

[TPS_SYST_01037] primitive Data Mapping of UINT8-Arrays ⌈ The primitive Data Mapping may also be used for the Data Mapping of UINT8-Arrays. This supports an optimized definition of the Data Mapping. ⌋

In other words it is allowed to map an array VariableDataPrototype consisting of UINT8 elements to exactly one SystemSignal in the context of one SenderReceiverToSignalMapping. A UINT8 element may be a String or an array that contains array elements of Integer type with range 0..255.

Background: In the ECU Configuration of the AUTOSAR COM module such a SystemSignal will be mapped to a COM Signal with the ComSignalType UINT8_N.

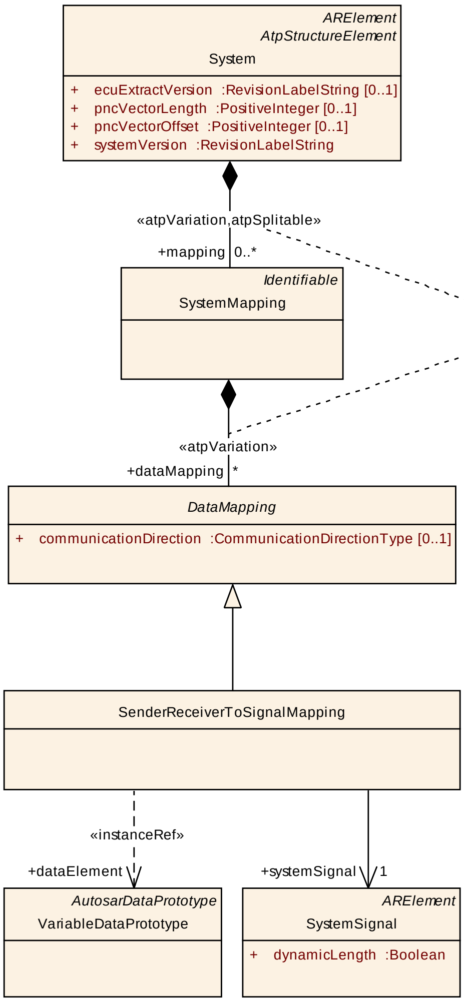

Figure 5.8: Mapping of data elements with primitive datatypes (SenderRecPrimitiveTypeMapping)

Class	SenderReceiverToSignalMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	Mapping of a sender receiver communication data element with a primitive datatype to a signal.			
Base	ARObject,DataMapping			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
dataElement	VariableDataPrototype	1	iref	Reference to the data element, which ought to be sent over the Communication bus.
systemSignal	SystemSignal	1	ref	Reference to the system signal used to carry the data element.

Table 5.11: SenderReceiverToSignalMapping

5.2.1.2 Mapping of Variable Data Prototypes with composite datatypes on Signal Groups (Sender-Receiver Communication)

This section describes the mapping of VariableDataPrototypes typed by composite data types to SystemSignals.

The RTE is required to treat AUTOSAR signals transmitted using sender-receiver communication consistently. The SystemSignalGroups are used for this purpose.

It is not possible to map a VariableDataPrototype typed by composite data type directly to one SystemSignal.

[constr_3506] Mapping of composite data type to SystemSignals in SystemSignalGroup ⌈ The elements of a composite data type shall be mapped to single SystemSignals which shall be members of one SystemSignalGroup. ⌋

There is one exception 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 section 5.2.1.1).

The VariableDataPrototype that is referenced by dataElement can be typed by an ApplicationDataType or by an ImplementationDataType. This type decides which reference is used within the SenderRecRecordElementMapping and SenderRecArrayElementMapping.

Composite VariableDataPrototypes may nest within other composite VariableDataPrototypes. Each element typed by a primitive data type of such nested composite VariableDataPrototypes shall be mapped to one SystemSignal.

The mapping between the SystemSignal and the VariableDataPrototype is provided in the SenderReceiverToSignalGroupMapping (see Figure 5.9).

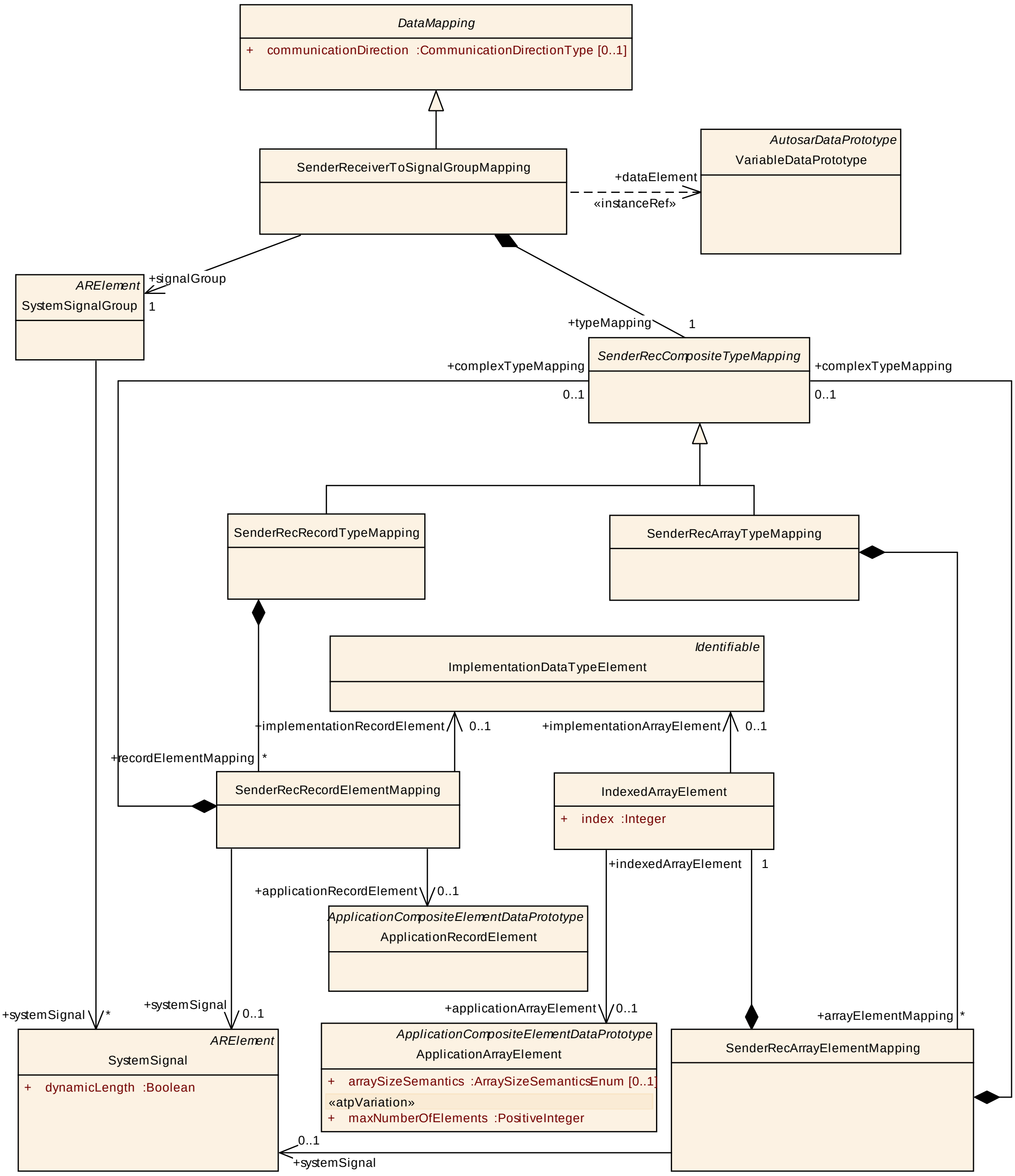

Figure 5.9: Mapping of data elements with composite data types (SenderRecCompositeTypeMapping)

[constr_3000] valid `SenderRecCompositeTypeMappings` ⌈`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 System-Signal. ⌋

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	**Datatype**	**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	SenderRecCompositeTypeMapping	1	aggr	The CompositeTypeMapping maps the the ApplicationArrayElements and ApplicationRecordElements to Signals of the SignalGroup.

Table 5.12: SenderReceiverToSignalGroupMapping

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 CompositeTypeMapping or aggregation between the RecordElementMapping and CompositeTypeMapping).			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
–	–	–	–	–

Table 5.13: SenderRecCompositeTypeMapping

Class	**SenderRecArrayTypeMapping**			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	If the ApplicationCompositeDataType is an Array, the "ArrayTypeMapping" will be used.			
Base	ARObject,SenderRecCompositeTypeMapping			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
arrayElementMapping	SenderRecArrayElementMapping	*	aggr	Each ApplicationArrayElement must be mapped on a SystemSignal.

Table 5.14: SenderRecArrayTypeMapping

Class	**SenderRecRecordTypeMapping**			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	If the ApplicationCompositeDataType is a Record, the "RecordTypeMapping" will be used.			
Base	ARObject,SenderRecCompositeTypeMapping			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
recordElementMapping	SenderRecRecordElementMapping	*	aggr	Each ApplicationRecordElement must be mapped on a SystemSignal.

Table 5.15: SenderRecRecordTypeMapping

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 applicationRecordElement shall be used. If the VariableDataPrototype is typed by the ImplementationDataType the reference implementationRecordElement shall be used. Either the implementationRecordElement or applicationRecordElement reference shall be used. If the element is composite, there will be no mapping to the SystemSignal (multiplicity 0). In this case the RecordElementMapping element will aggregate the complexTypeMapping element. In that way also the composite datatypes can be mapped to SystemSignals.			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
applicationRecordElement	ApplicationRecordElement	0..1	ref	Reference to an ApplicationRecordElement in the context of the dataElement or in the context of a composite element. This reference shall only be used if the VariableDataPrototype that is referenced by the SenderReceiverToSignalGroupMapping.dataElement is typed by an ApplicationDataType.
complexTypeMapping	SenderRecCompositeTypeMapping	0..1	aggr	This aggregation will be used if the element is composite.

Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
implementationRecordElement	ImplementationDataTypeElement	0..1	ref	Reference to an ImplementationRecordElement in the context of the dataElement or in the context of a composite element. This reference shall only be used if VariableDataPrototype that is referenced by the SenderReceiverToSignalGroupMapping.dataElement is typed by an ImplementationDataType.
systemSignal	SystemSignal	0..1	ref	Reference to the system signal used to carry the primitive ApplicationRecordElement.

Table 5.16: SenderRecRecordElementMapping

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	**Datatype**	**Mul.**	**Kind**	**Note**
complexTypeMapping	SenderRecCompositeTypeMapping	0..1	aggr	This aggregation will be used if the element is composite.
indexedArrayElement	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 5.17: SenderRecArrayElementMapping

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	**Datatype**	**Mul.**	**Kind**	**Note**
applicationArrayElement	ApplicationArrayElement	0..1	ref	Reference to an ApplicationArrayElement in an array. This reference shall only be used if the referenced context element (VariableDataPrototype that is referenced by the SenderReceiverToSignalGroupMapping or ArgumentDataPrototype that is referenced by the ClientServerCompositeTypeMapping) is typed by an ApplicationDataType.
implementationArrayElement	ImplementationDataTypeElement	0..1	ref	Reference to an ImplementationDataTypeElement in an array. This reference shall only be used if the referenced context element (VariableDataPrototype that is referenced by the SenderReceiverToSignalGroupMapping or ArgumentDataPrototype that is referenced by the ClientServerCompositeTypeMapping) is typed by an ImplementationDataType.
index	Integer	1	attr	Position of an element in an array.

Table 5.18: IndexedArrayElement

Figure 5.10 shows a mapping example for nested composite data types.

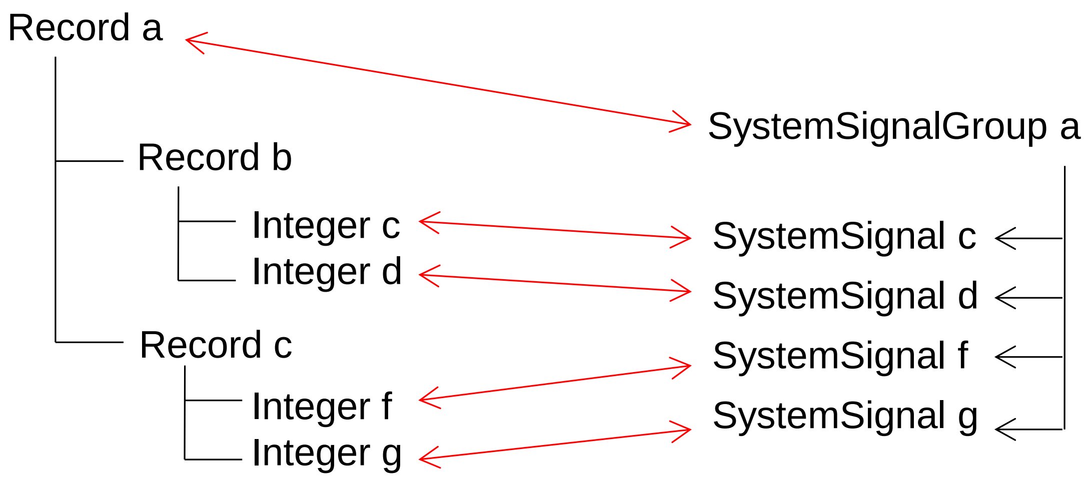

Figure 5.10: Mapping example for nested composite data types

Record a is mapped with SenderReceiverToSignalGroupMapping to a SystemSignalGroup. The content of Record a is mapped with the SenderRecRecordTypeMapping. Since the first element of *Record a* is *Record b* the SenderRecRecordElementMapping does not contain a reference to a SystemSignal because signals apply only to atomic data items. Instead it contains a complexTypeMapping with two SenderRecRecordElementMappings for *Integer c* and *Integer d*. These two elements are mapped to SystemSignals.

5.2.1.3 Mapping of Client Server Operations to System Signals

This section describes the mapping of ClientServerOperations to SystemSignals (see Figure 5.11).

[TPS_SYST_01148] Mapping of IN and INOUT ArgumentDataPrototypes to callSignals ⌈ The ArgumentDataPrototypes that are passed to the operation (i.e. the direction is "in") and the ArgumentDataPrototypes that are passed to and returned from the operation (i.e. the direction is "inout") are expected to be mapped to the callSignal by the serializer. ⌋

[TPS_SYST_01149] Mapping of OUT and INOUT ArgumentDataPrototypes to returnSignals ⌈ The ArgumentDataPrototypes that are returned from the operation (i.e. the direction is "out") and the ArgumentDataPrototypes that are passed to and returned from the operation (i.e. the direction is "inout") are expected to be mapped to the returnSignal by the serializer. ⌋

[TPS_SYST_01150] Mapping of returnSignal and callSignal to COM Signal ⌈ In the ECU Configuration of the AUTOSAR COM module the returnSignal and the callSignal are expected to be mapped to COM Signals with the ComSignalType UINT8_N or UINT8_DYN. ⌋

The ClientServerToSignalMapping contains the necessary information how data shall be serialized. The referenced SerializationTechnology contains the necessary information which serialization algorithm shall be applied. The implementation of this algorithm is provided via a BSW module.

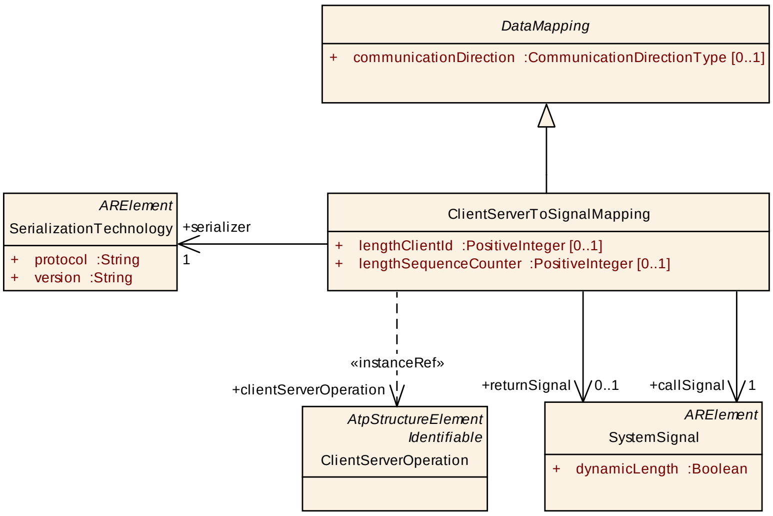

Figure 5.11: Mapping of a ClientServerOperation to a callSignal and a returnSignal

Class	**ClientServerToSignalMapping**			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	This element maps the ClientServerOperation to call- and return-SystemSignals. The serialization is defined by the referenced SerializationTechnology. **Tags:** atp.Status=draft			
Base	ARObject,DataMapping			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
callSignal	SystemSignal	1	ref	Reference to the callSignal to which the IN and INOUT ArgumentDataPrototypes are mapped.
clientServerOperation	ClientServerOperation	1	iref	Reference to a ClientServerOperation, which is mapped to a call SystemSignal and a return SystemSignal.
lengthClientId	PositiveInteger	0..1	attr	This attribute defines the length of the used client identifier in bits. If the attribute does not exist or its value is set to 0 this means that the client identifier is not used.
lengthSequenceCounter	PositiveInteger	0..1	attr	The purpose of a sequence counter is to map a response to the correct request of a known client. This attribute describes the length of the used sequence counter in bits. If the attribute does not exist or its value is set to 0 this means that the sequence counter is not used.
returnSignal	SystemSignal	0..1	ref	Reference to the returnSignal to which the OUT and INOUT ArgumentDataPrototypes are mapped.
serializer	SerializationTechnology	1	ref	The referenced SerializationTechnology element contains the necessary information how data shall be serialized.

Attribute	Datatype	Mul.	Kind	Note

Table 5.19: ClientServerToSignalMapping

Class	**SerializationTechnology**			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	Identifies the used serialization technology. The corresponding serialization plugin has to be provided on each affected ECU by the integrator. **Tags:** atp.Status=draft; atp.recommendedPackage=SerializationTechnologies			
Base	ARElement,ARObject,CollectableElement,Identifiable,MultilanguageReferrable,PackageableElement,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
protocol	String	1	attr	Payload serialization technology.
version	String	1	attr	Version of the used serialization technology.

Table 5.20: SerializationTechnology

5.2.1.4 Mapping of Client Server Operations to Signal Groups

This chapter is deprecated and will be removed in the future. Please note that the ClientServerToSignalGroupMapping is no longer supported by the RTE.

The Client/Server interfaces aggregate a number of Client Server operations. Each description of an operation consists of the description of its arguments. Furthermore, the RTE is responsible to map a response to the corresponding request. For this mapping transaction handles are used. The transaction handle contain a client identifier and a sequence counter.

The arguments, application errors, client identifier and sequence counter of an operation are mapped to SystemSignals of two dedicated SystemSignalGroup elements; one for the request and one for the response. The RTE Client Server Protocol is used to provide a specific semantics to each of these SystemSignalGroups and SystemSignals, also those which are introduced only to support the protocol. This is described in more detail in [16].

The relationship between the SystemSignals and the ArgumentDataPrototypes is provided in the ClientServerToSignalGroupMapping (see Figure 5.12) [2].

[constr_1208] Existence of the attribute DataMapping.communicationDirection in the context of a ClientServerInterface ⌈ The following conditions shall be fulfilled regarding the existence and values of the attribute DataMapping.communicationDirection that refers to a PortPrototype typed by a ClientServerInterface as the context PortPrototype:

[2] Please note that the ClientServerToSignalGroupMapping is deprecated and is replaced by the ClientServerToSignalMapping.

- If the DataMapping refers to a PRPortPrototype as the context PortPrototype the attribute DataMapping.communicationDirection shall exist.

  The value of the attribute DataMapping.communicationDirection shall be set according to the role taken by the PRPortPrototype. This means that [constr_1208] shall apply in terms of the regulations for PPortPrototype because the PRPortPrototype can **only** act as a server.

- 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 depending on the value of the attribute ArgumentDataPrototype.attribute:

  - If the value of ArgumentDataPrototype.attribute is set to in the value of DataMapping.communicationDirection shall be set to in.
  - If the value of ArgumentDataPrototype.attribute is set to out the value of DataMapping.communicationDirection shall be set to out.
  - If the value of ArgumentDataPrototype.attribute is set to inout two separate ClientServerPrimitiveTypeMapping or ClientServerCompositeTypeMapping (depending on the data type used to type the applicable ArgumentDataPrototype) shall exist where one has the attribute DataMapping.communicationDirection set to in and the other one has the attribute DataMapping.communicationDirection 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 depending on the value of the attribute ArgumentDataPrototype.attribute:

  - If the value of ArgumentDataPrototype.attribute is set to in the value of DataMapping.communicationDirection shall be set to out.
  - If the value of ArgumentDataPrototype.attribute is set to out the value of DataMapping.communicationDirection shall be set to in.
  - If the value of ArgumentDataPrototype.attribute is set to inout two separate ClientServerPrimitiveTypeMapping or ClientServerCompositeTypeMapping (depending on the data type used to type the applicable ArgumentDataPrototype) shall exist where one has the attribute DataMapping.communicationDirection set to in and the other one has the attribute DataMapping.communicationDirection set to out.

⌋

5.2.1.4.1 Primitive Argument Mapping

[TPS_SYST_01038] Mapping of primitive arguments ⌈ Each primitive argument shall be mapped directly onto one SystemSignal by ClientServerPrimitiveTypeMapping. ⌋

[constr_3033] Criteria for primitive argument mapping ⌈ The ArgumentDataPrototype referenced by argument shall be typed by one of

- ApplicationPrimitiveDataType of category VALUE, BOOLEAN, and STRING and for which a DataTypeMappingSet exists that points to an ImplementationDataType that fulfills all of the following conditions:
  - The ImplementationDataType is either
    - of category TYPE_REFERENCE that eventually references an ImplementationDataType of category VALUE **or**
    - the ImplementationDataType is of category VALUE.
  - The ImplementationDataType either
    - represents the platform type uint8 **or**
    - references a SwBaseType with a SwBaseType.baseTypeDefinition.baseTypeSize set to value 8 and the SwBaseType.baseTypeDefinition.baseTypeEncoding set to NONE.
- ImplementationDataType of category ARRAY that has a subElement that fulfills all of the following conditions:
  - the subElement is either
    - of category TYPE_REFERENCE that (by reference to a swDataDefProps.implementationDataType) eventually references an ImplementationDataType of category VALUE **or**
    - the subElement is of category VALUE.
  - the subElement (by reference to a swDataDefProps.implementationDataType) either
    - implements the platform type uint8 **or**
    - references a SwBaseType with a SwBaseType.baseTypeDefinition.baseTypeSize set to value 8 and the SwBaseType.baseTypeDefinition.baseTypeEncoding set to NONE.
- ApplicationArrayDataType for which a DataTypeMap exists that points to an ImplementationDataType that fulfills the above mentioned condition.

Alternatively, the following rules apply for a scenario where a DataTypeMap does not yet exist:

The ArgumentDataPrototype referenced by argument shall be typed by one of

- ApplicationPrimitiveDataType of category BOOLEAN
- ApplicationPrimitiveDataType of category VALUE if the following conditions are fulfilled:
  - ApplicationPrimitiveDataType.swDataDefProps.dataConstr exists and refers to a PhysConstrs.
  - ApplicationPrimitiveDataType.swDataDefProps.compuMethod exists and refers to a CompuMethod of category TEXTTABLE and CompuMethod.compuPhysToInternal exists.
  - Application of ApplicationPrimitiveDataType.swDataDefProps.compuMethod to ApplicationPrimitiveDataType.swDataDefProps.dataConstr yields a numerical range in [0 .. 255].
- ApplicationPrimitiveDataType of category STRING if
  - ApplicationPrimitiveDataType.swDataDefProps.swRecordLayout exists and values of SwRecordLayout.swRecordLayoutGroup.swRecordLayoutGroupFrom and SwRecordLayout.swRecordLayoutGroup.swRecordLayoutGroupTo are both set to 1.
  - ApplicationPrimitiveDataType.swDataDefProps.swTextProps exists and refers to an SwBaseType where the SwBaseType.baseTypeDefinition.baseTypeEncoding is set to NONE and the value of SwBaseType.baseTypeDefinition.baseTypeSize is set to 8.
- ApplicationArrayDataType where the aggregated element fulfills the following conditions:
  - ApplicationPrimitiveDataType.swDataDefProps.dataConstr exists and refers to a PhysConstrs.
  - ApplicationPrimitiveDataType.swDataDefProps.compuMethod exists and refers to a CompuMethod of category TEXTTABLE and CompuMethod.compuPhysToInternal exists.
  - Application of ApplicationPrimitiveDataType.swDataDefProps.compuMethod to ApplicationPrimitiveDataType.swDataDefProps.dataConstr yields a numerical range in [0 .. 255].

⌋

[TPS_SYST_01039] primitive Argument Mapping of UINT8-Arrays ⌈ The primitive Argument Mapping may also be used for the Argument Mapping of UINT8-Arrays. ⌋

In other words it is allowed to map an array ArgumentDataPrototype consisting of UINT8 elements to exactly one SystemSignal via the ClientServerPrimitiveTypeMapping. A UINT8 element may be a String or an array that contains array elements of Integer type with range 0..255.

Background: In the ECU Configuration of the AUTOSAR COM module such a SystemSignal will be mapped to a COM Signal with the ComSignalType UINT8_N.

5.2.1.4.2 Composite Argument Mapping

[TPS_SYST_01040] Mapping of composite arguments ⌈ Each composite argument shall be decomposed into single SystemSignals by ClientServerCompositeTypeMapping. ⌋

The ArgumentDataPrototype that is referenced by ClientServerCompositeTypeMapping can be typed by an ApplicationDataType or by an ImplementationDataType. This type decides which reference is used within the ClientServerRecordElementMapping and ClientServerArrayElementMapping.

In a complete System with category SYSTEM_DESCRIPTION, it is sufficient to refer to the operation in the PPortPrototype to define the mapping of the communication between a provider and its receivers. This is possible since the connectors implicitly define which RPortPrototypes are connected to the PPortPrototype. In an System with category SYSTEM_EXTRACT or ECU_EXTRACT, where only the relevant parts of the SW compositions are defined, it is in some cases also necessary to refer to RPortPrototypes, if the corresponding PPortPrototype is not part of the extract. This is described in more details in chapter 9.2 for the System Extract and chapter 10.2.3 for the ECU Extract.

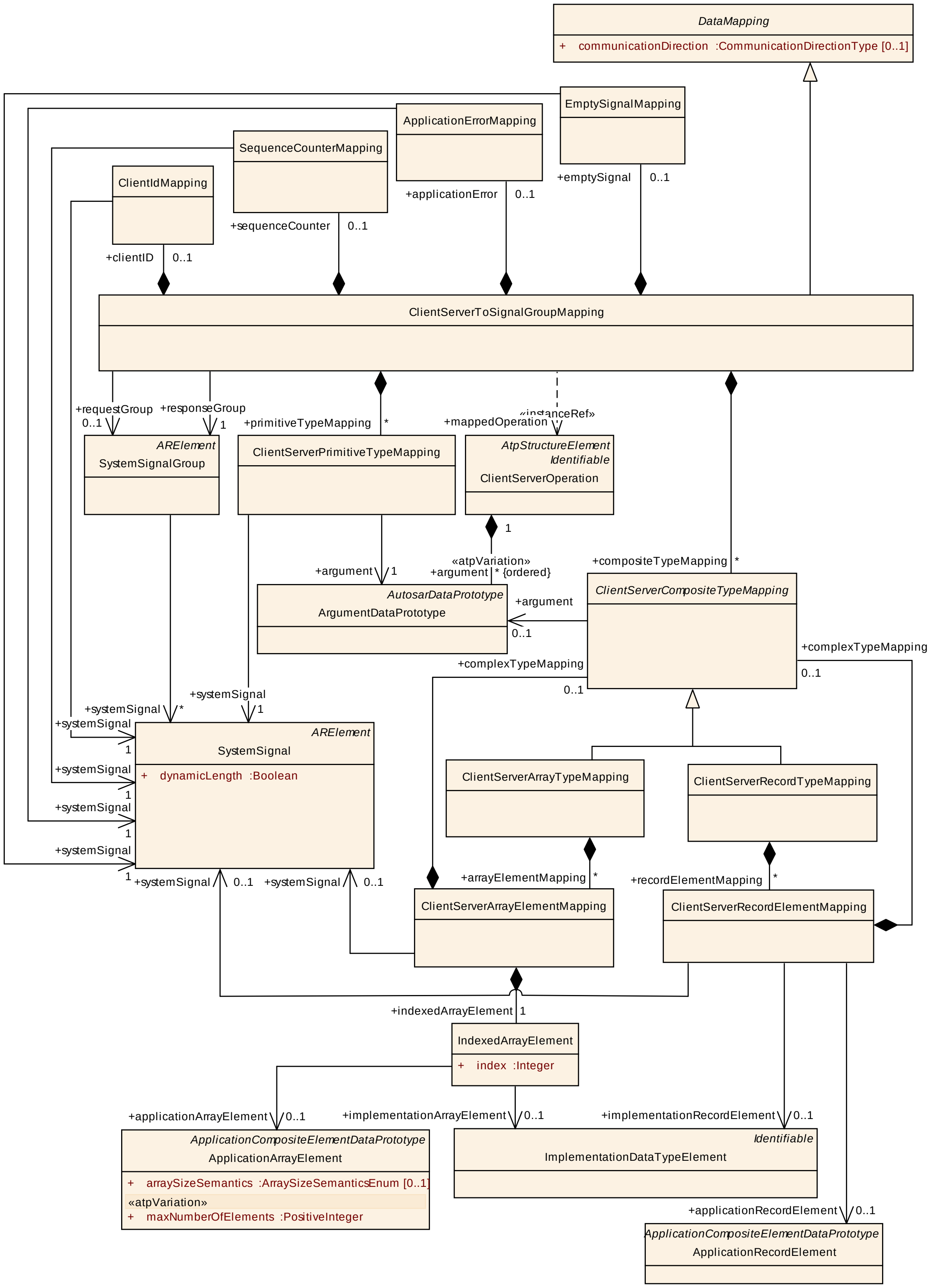

Figure 5.12: Operation Mapping (ClientServerOperationMapping)

[constr_3001] valid ClientServerToSignalGroupMappings ⌈ SystemSignals that are referenced by a ClientServerArrayTypeMapping or

ClientServerRecordTypeMapping within the context of ClientServerToSignalGroupMapping shall also be referenced by ClientServerToSignalGroupMapping.requestGroup.systemSignal or ClientServerToSignalGroupMapping.responseGroup.systemSignal. ⌋

Class	**ClientServerToSignalGroupMapping**			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	This mapping is deprecated and will be removed in future. It is replaced by the ClientServerToSignalMapping. Old description: Mapping of client server operation arguments to signals of a signal group. Arguments with a primitive datatype will be mapped via the "ClientServerPrimitiveTypeMapping" element. Arguments with composite datatypes will be mapped via the "CompositeTypeMapping" element. **Tags:** atp.Status=obsolete			
Base	ARObject,DataMapping			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
applicationError	ApplicationErrorMapping	0..1	aggr	In client server communication, the server may return any value within the application error range.
clientID	ClientIdMapping	0..1	aggr	In case of a server on one ECU with multiple clients on other ECUs, the client server communication shall use different unique COM signals and signal groups for each client to allow the identification of the client associated with each system signal.
compositeTypeMapping	ClientServerCompositeTypeMapping	*	aggr	Mapping of arguments with composite datatypes.
emptySignal	EmptySignalMapping	0..1	aggr	An emptySignal is created if no actual data is configured for a client-server communication, but if the RTE shall send a SignalGroup to initiate the communication. An EmptySignalMapping shall only reference a SystemSignal that is referenced by an ISignal with length equal to zero.
mappedOperation	ClientServerOperation	1	iref	Reference to a Operation, which is mapped to a signal group.
primitiveTypeMapping	ClientServerPrimitiveTypeMapping	*	aggr	Mapping of an argument with a primitive datatype to a signal.
requestGroup	SystemSignalGroup	0..1	ref	Reference to the signal group which contains the references to request signals used to transport the IN and INOUT arguments of the operation.
responseGroup	SystemSignalGroup	1	ref	Reference to the signal group which contains the references to response signals used to transport the OUT and INOUT arguments of the operation.
sequenceCounter	SequenceCounterMapping	0..1	aggr	The purpose of sequence counters is to map a response to the correct request of a known client.

Table 5.21: ClientServerToSignalGroupMapping

[constr_3026] valid EmptySignalMappings ⌈ An EmptySignalMapping shall only reference a SystemSignal that is referenced by an ISignal with length equal to zero. ⌋

Class	**ClientIdMapping**			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	In case of a server on one ECU with multiple clients on other ECUs, the client server communication shall use different unique COM signals and signal groups for each client to allow the identification of the client associated with each system signal. The ClientId is mapped to the requestGroup and to the responseGroup.			
Base	ARObject			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
systemSignal	SystemSignal	1	ref	Reference to the SystemSignal with the ClientID.

Table 5.22: ClientIdMapping

Class	**SequenceCounterMapping**			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	The purpose of sequence counters is to map a response to the correct request of a known client. The SequenceCounter is mapped to the requestGroup and to the responseGroup.			
Base	ARObject			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
systemSignal	SystemSignal	1	ref	Reference to the SystemSignal with the SequenceCounter.

Table 5.23: SequenceCounterMapping

Class	**ApplicationErrorMapping**			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	In client server communication, the server may return any value within the application error range. The ApplicationError is mapped to the responseGroup.			
Base	ARObject			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
systemSignal	SystemSignal	1	ref	Reference to the SystemSignal with the ApplicationError.

Table 5.24: ApplicationErrorMapping

Class	EmptySignalMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	If no actual data is configured for a client server communication the element EmptySignalMapping shall be used. An EmptySignalMapping shall only reference a SystemSignal that is referenced by an ISignal with length equal to zero. In this case there shall be an "update-bit" configured. The EmptySignal can be mapped to the response group or to request group.			
Base	ARObject			
Attribute	Datatype	Mul.	Kind	Note
systemSignal	SystemSignal	1	ref	Reference to a SystemSignal with "signalLength" = 0 and an UpdateBit.

Table 5.25: EmptySignalMapping

Class	ClientServerPrimitiveTypeMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	Mapping of an argument with a primitive datatype to a signal.			
Base	ARObject			
Attribute	Datatype	Mul.	Kind	Note
argument	ArgumentDataPrototype	1	ref	Reference to an argument in the context of the mappedOperation.
systemSignal	SystemSignal	1	ref	Reference to the system signal used to carry the argument

Table 5.26: ClientServerPrimitiveTypeMapping

Class	ClientServerCompositeTypeMapping (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 CompositeTypeMapping or aggregation between the RecordElementMapping and CompositeTypeMapping).			
Base	ARObject			
Attribute	Datatype	Mul.	Kind	Note
argument	ArgumentDataPrototype	0..1	ref	Reference to an argument in the context of the mappedOperation. Only ClientServerCompositeTypeMapping elements that are directly aggregated by the ClientServerToSignalGroupMapping shall contain this reference.

Table 5.27: ClientServerCompositeTypeMapping

Class	**ClientServerArrayTypeMapping**			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	If the ApplicationCompositeDataType is an Array, the "ArrayTypeMapping" will be used.			
Base	ARObject,ClientServerCompositeTypeMapping			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
arrayElementMapping	ClientServerArrayElementMapping	*	aggr	Each ApplicationArrayElement must be mapped on a SystemSignal.

Table 5.28: ClientServerArrayTypeMapping

Class	**ClientServerArrayElementMapping**			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	The ApplicationArrayElement 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 ArgumentDataPrototype that is referenced by ClientServerCompositeTypeMapping 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 array element always needs to be specified.			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
complexTypeMapping	ClientServerCompositeTypeMapping	0..1	aggr	This aggregation will be used if the element is composite.
indexedArrayElement	IndexedArrayElement	1	aggr	Reference to an indexed array element in the context of the mappedOperation 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 5.29: ClientServerArrayElementMapping

Class	**ClientServerRecordTypeMapping**			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	If the ApplicationCompositeDataType is a Record, the "RecordTypeMapping" will be used.			
Base	ARObject,ClientServerCompositeTypeMapping			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
recordElementMapping	ClientServerRecordElementMapping	*	aggr	Each ApplicationRecordElement must be mapped on a SystemSignal.

Table 5.30: ClientServerRecordTypeMapping

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	*Datatype*	*Mul.*	*Kind*	*Note*
applicationArrayElement	ApplicationArrayElement	0..1	ref	Reference to an ApplicationArrayElement in an array. This reference shall only be used if the referenced context element (VariableDataPrototype that is referenced by the SenderReceiverToSignalGroupMapping or ArgumentDataPrototype that is referenced by the ClientServerCompositeTypeMapping) is typed by an ApplicationDataType.
implementationArrayElement	ImplementationDataTypeElement	0..1	ref	Reference to an ImplementationDataTypeElement in an array. This reference shall only be used if the referenced context element (VariableDataPrototype that is referenced by the SenderReceiverToSignalGroupMapping or ArgumentDataPrototype that is referenced by the ClientServerCompositeTypeMapping) is typed by an ImplementationDataType.
index	Integer	1	attr	Position of an element in an array.

Table 5.31: IndexedArrayElement

Class	**ClientServerRecordElementMapping**			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	Mapping of a primitive record element to a SystemSignal. If the ArgumentDataPrototype that is referenced by ClientServerCompositeTypeMapping is typed by an ApplicationDataType the reference to the ApplicationRecordElement shall be used. If the VariableDataPrototype is typed by the ImplementationDataType the reference to the ImplementationRecordElement shall be used. If the element is composite, there will be no mapping (multiplicity 0). In this case the "RecordElementMapping" Element will aggregate the "TypeMapping" Element. In that way also the composite datatypes can be mapped to SystemSignals. Regardless whether composite or primitive record element is mapped the record element always needs to be specified.			
Base	ARObject			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
applicationRecordElement	ApplicationRecordElement	0..1	ref	Reference to a applicationRecordElement in the context of the mappedOperation or in the context of a composite element. This reference shall only be used if the ArgumentDataPrototype that is referenced by the ClientServerCompositeTypeMapping is typed by an ApplicationDataType.
complexTypeMapping	ClientServerCompositeTypeMapping	0..1	aggr	This aggregation will be used if the element is composite.
implementationRecordElement	ImplementationDataTypeElement	0..1	ref	Reference to a ImplementationRecordElement in the context of the mappedOperation or in the context of a composite element. This reference shall only be used if the ArgumentDataPrototype that is referenced by the ClientServerCompositeTypeMapping is typed by an ImplementationDataType.
systemSignal	SystemSignal	0..1	ref	Reference to the system signal used to carry the primitive ApplicationRecordElement.

Table 5.32: ClientServerRecordElementMapping

5.2.1.5 Mapping of a ApplicationCompositeElementDataPrototype within a composite application data type on a System Signal (Sender-Receiver Communication)

SenderReceiverCompositeElementToSignalMapping is used to map a ApplicationCompositeElementDataPrototype that is aggregated within a composite data type (record element or an array element) to a SystemSignal.

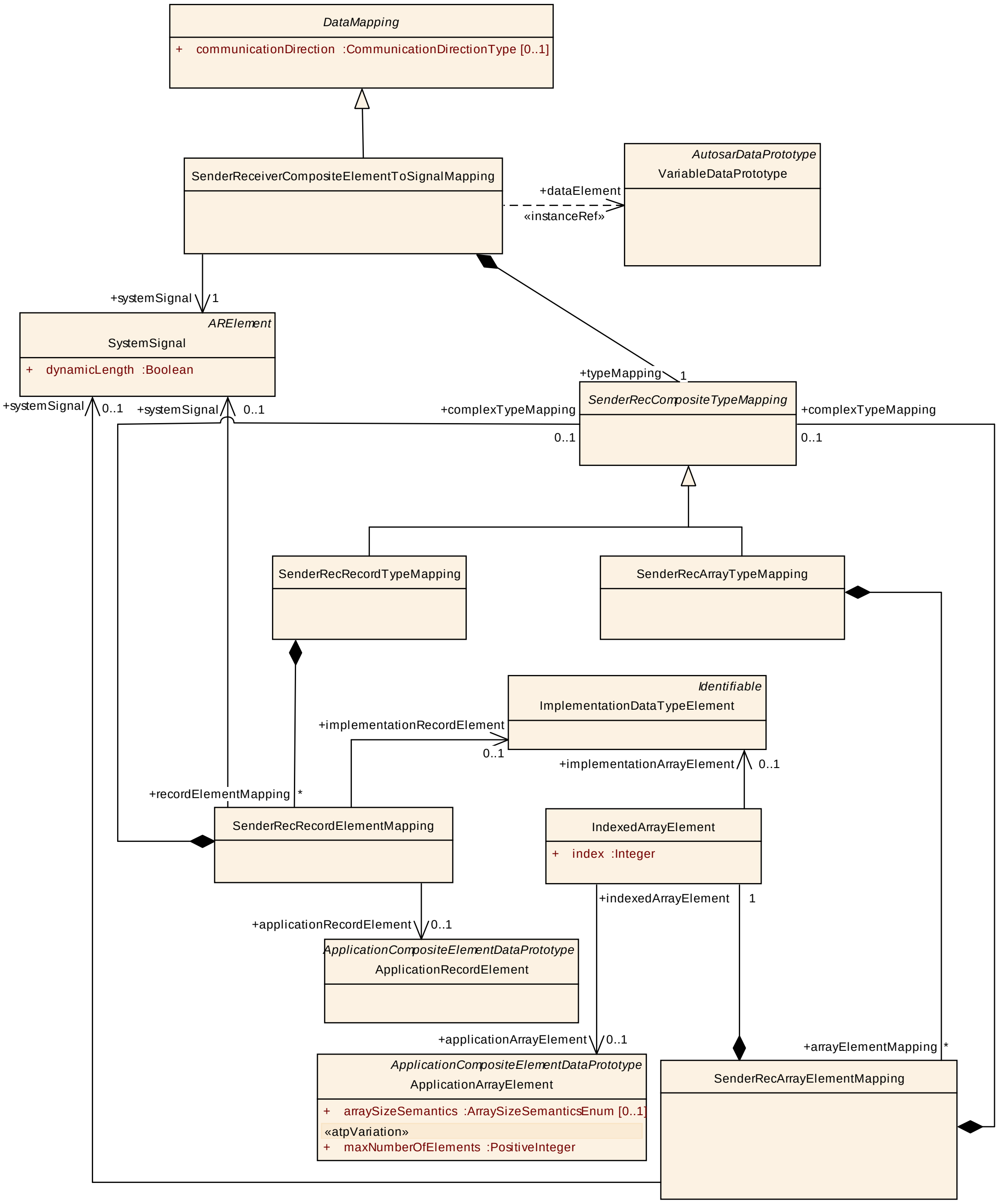

Figure 5.13: Mapping of a Variable Data Prototype which is aggregated within a composite data type on a System Signal

[constr_3058] References from SenderRecArrayElementMapping and from SenderRecRecordElementMapping to SystemSignals 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 a ApplicationCompositeDataType (ApplicationCompositeDataType.element) that types a dataElement in a RPortPrototype or PRPortPrototype in its receiver role a DataMapping exists. ⌋

[TPS_SYST_01143] DataMapping on the sender side for elements of a composite data type ⌈ On the sender side, it is possible that only a subset of elements of a ApplicationCompositeElementDataPrototypes a dataElement in a PPortPrototype or a PRPortPrototype in its sender role is referenced by a DataMapping. ⌋

Class	**SenderReceiverCompositeElementToSignalMapping**			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	Mapping of an Variable Data Prototype which is aggregated within a composite datatype to a SystemSignal (only one element of the composite data type is mapped).			
Base	ARObject,DataMapping			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
dataElement	VariableDataPrototype	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.
typeMapping	SenderRecCompositeTypeMapping	1	aggr	The CompositeTypeMapping maps one VariableDataPrototype of the composite data type to a SystemSignal.

Table 5.33: SenderReceiverCompositeElementToSignalMapping

SenderRecCompositeTypeMapping and all subclasses are described in section 5.2.1.2

5.2.1.6 Mapping of Trigger to SystemSignal

[TPS_SYST_05001] Send a Trigger across a network ⌈ In order to be able to send a Trigger across a network to trigger a RunnableEntity deployed to a different EcuInstance it is possible to define a TriggerToSignalMapping that maps a Trigger to a SystemSignal in the role systemSignal. ⌋

[constr_1198] TriggerToSignalMapping.systemSignals 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] ISignals relating to systemSignals 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. ⌋

[TPS_SYST_05002] The value of startPosition is irrelevant ⌈ The value of startPosition shall not be considered inside an ISignalToIPduMapping that references an ISignal used to reference a TriggerToSignalMapping.systemSignal that in turn is referenced by a TriggerToSignalMapping. ⌋

[constr_3065] Mapping of queued Triggers to SystemSignals is prohibited ⌈ A TriggerToSignalMapping of a Trigger with swImplPolicy set to queued is prohibited. ⌋

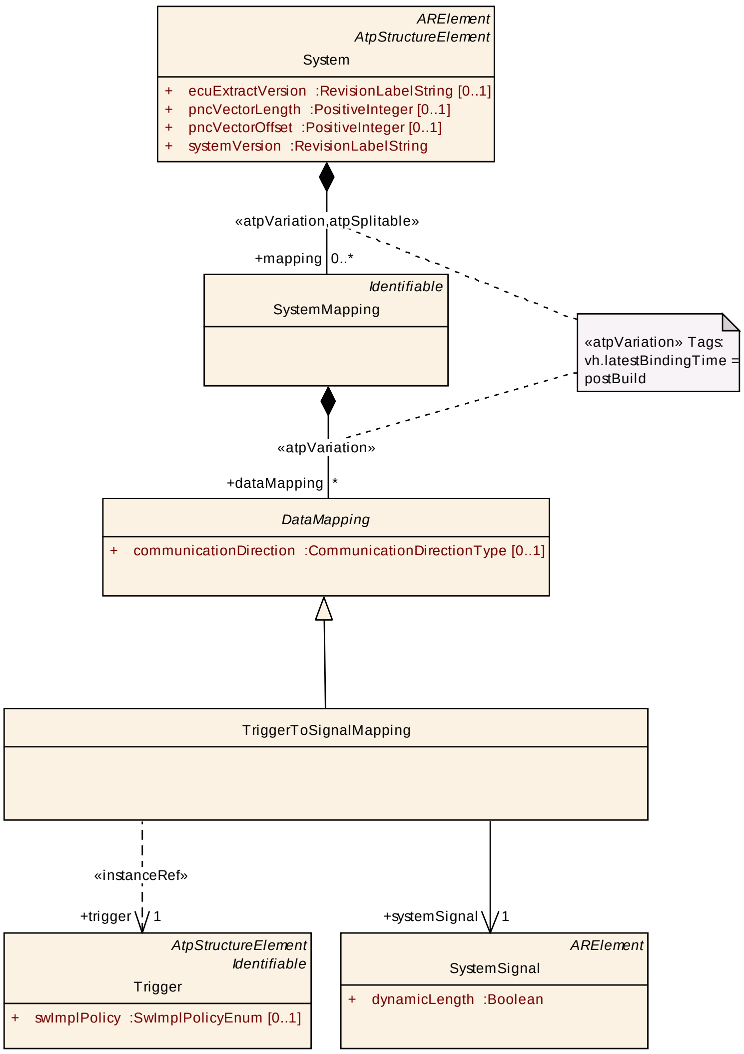

Figure 5.14: Structure of a TriggerToSignalMapping

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	**Datatype**	**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 5.34: TriggerToSignalMapping

5.2.2 Signal Path Constraint

One task of the System Generator is to define the needed communication infrastructure (e.g. ISignals, Pdus, Frames) between ECUs. The System Generator often has the choice between alternative paths through the topology. In the example shown in Figure 5.15 the System Generator would have the choice between two paths (Path1: CAN3 or Path2: CAN1-GW-CAN2) for a signal that is send by ECU2 and is received by ECU4. If no further information is given the decision will be made e.g. by means of boundary conditions like busload, transmissions speed, etc.

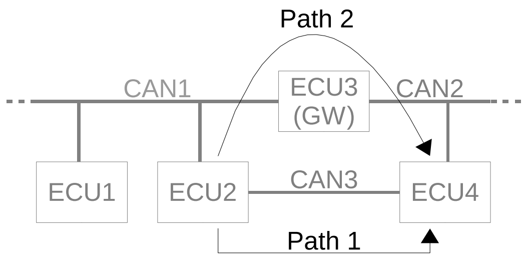

Figure 5.15: Example for a Communication Path

Signal Mapping Constraints allow to further restrict or specify the path(s) a signal is allowed to be transmitted over. A path is specified by an list of PhysicalChannels.

There exist four different constraints for signals regarding the signal path (see Figure 5.16):

[TPS_SYST_01041] CommonSignalPath definition ⌈ The CommonSignalPath describes that two or more signals shall take the same path in the topology. ⌋*(RS_SYST_00017)*

[TPS_SYST_01042] ForbiddenSignalPath definition ⌈ The ForbiddenSignalPath describes the path that one or more signals shall not take in the topology, e.g. in case of safety critical transmission. ⌋*(RS_SYST_00020)*

[TPS_SYST_01043] PermissibleSignalPath definition ⌈ The PermissibleSignalPath describes the path one or more signals may take in the topology. If more than one PermissibleSignalPath is defined for the same signal/operation attributes, any of them may be chosen. ⌋*(RS_SYST_00019, RS_SYST_00016)*

[TPS_SYST_01044] SeparateSignalPath definition ⌈ The SeparateSignalPath describes that two or more signals shall take separate paths in the topology e.g. in case of redundant transmission. ⌋*(RS_SYST_00018)*

It is also possible that the same signal is aggregated two times by the SeparateSignalPath element to indicate that this signal should be transmitted redundantly over two different paths.

The meta-model part, which describes the Communication Path constraints, will be explained in the following sections.

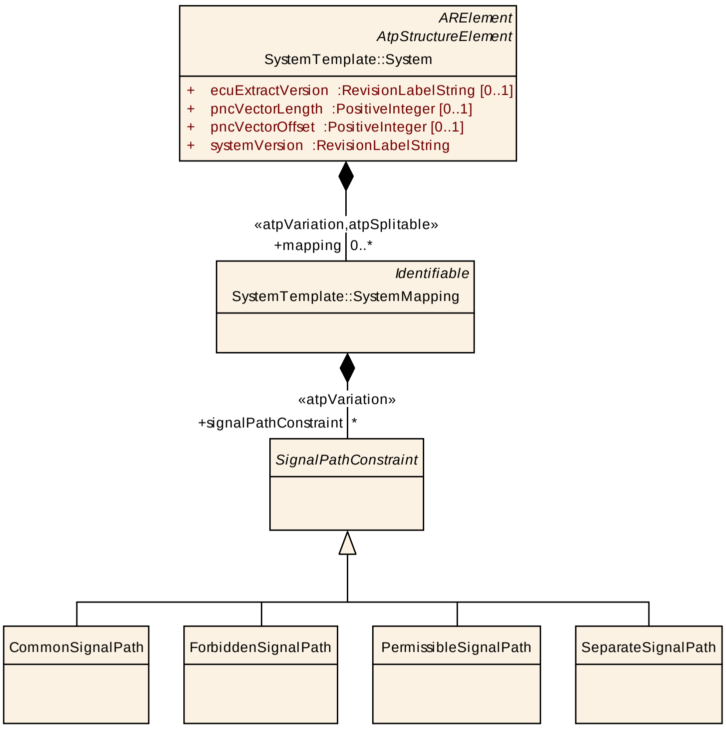

Figure 5.16: Communication Path Description (SignalPathConstraints)

5.2.2.1 CommonSignalPath

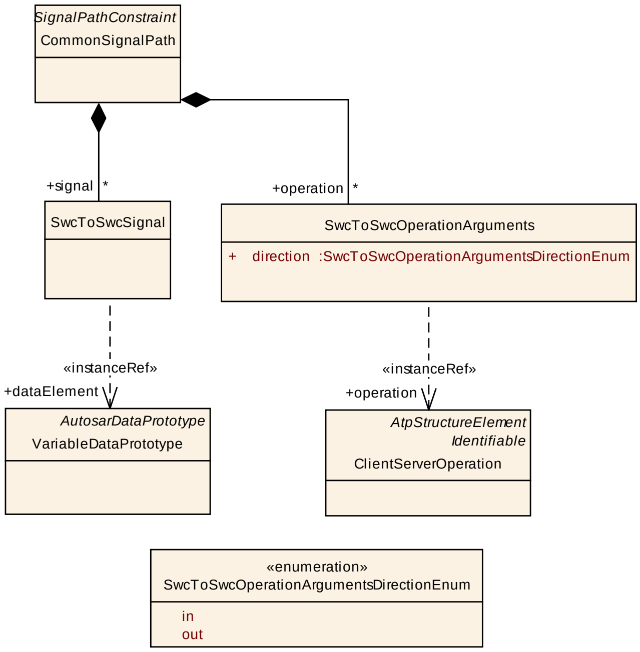

Figure 5.17: Description of signals that must take the same way in the topology (CommonSignalPath)

Class	**CommonSignalPath**			
Package	M2::AUTOSARTemplates::SystemTemplate::SignalPaths			
Note	The CommonSignalPath describes that two or more SwcToSwcSignals and/or SwcToSwcOperationArguments must take the same way (Signal Path) in the topology.			
Base	ARObject,SignalPathConstraint			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
operation	SwcToSwcOperationArguments	*	aggr	
signal	SwcToSwcSignal	*	aggr	The SwcToSwcSignals that must take the same way (Signal Path) in the topology.

Table 5.35: CommonSignalPath

Class	**SwcToSwcSignal**			
Package	M2::AUTOSARTemplates::SystemTemplate::SignalPaths			
Note	The SwcToSwcSignal describes the information (data element) that is exchanged between two SW Components. On the SWC Level it is possible that a SW Component sends one data element from one P-Port to two different SW Components (1:n Communication). The SwcToSwcSignal describes exactly the information which is exchanged between one P-Port of a SW Component and one R-Port of another SW Component.			
Base	ARObject			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
dataElement	VariableDataPrototype	2	iref	Reference to a data element on the PPortPrototype and to the same data element on the RPortPrototype.

Table 5.36: SwcToSwcSignal

Class	**SwcToSwcOperationArguments**			
Package	M2::AUTOSARTemplates::SystemTemplate::SignalPaths			
Note	The SwcToSwcOperationArguments describes the information (client server operation arguments, plus the operation identification, if required) that are exchanged between two SW Components from exactly one client to one server, or from one server back to one client. The direction attribute defines which direction is described. If direction == IN, all arguments sent from the client to the server are described by the SwcToSwcOperationArguments, in direction == OUT, it's the arguments sent back from server to client.			
Base	ARObject			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
direction	SwcToSwcOperationArgumentsDirectionEnum	1	attr	Direction addressed by this SwcToSwcClientServerOperation element.
operation	ClientServerOperation	2	iref	Reference to the operation at the client and at the server side whose arguments are described by SwcToSwcOperationArguments. The two ports referenced must be connected by a connector in the software component description.

Table 5.37: SwcToSwcOperationArguments

Enumeration	**SwcToSwcOperationArgumentsDirectionEnum**
Package	M2::AUTOSARTemplates::SystemTemplate::SignalPaths
Note	Direction addressed by this element.
Literal	*Description*
in	IN (all IN and INOUT arguments)
out	OUT (all OUT and INOUT arguments) .

Table 5.38: SwcToSwcOperationArgumentsDirectionEnum

5.2.2.2 ForbiddenSignalPath

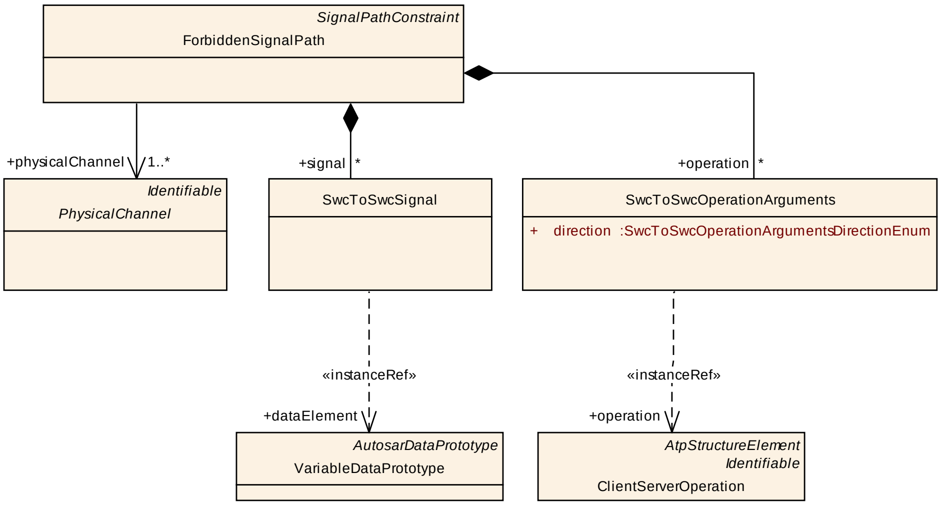

Figure 5.18: Description of the signal path that a signal must not take in the topology (ForbiddenSignalPath)

Class	**ForbiddenSignalPath**			
Package	M2::AUTOSARTemplates::SystemTemplate::SignalPaths			
Note	The ForbiddenSignalPath describes the physical channels which an element must not take in the topology. Such a signal path can be a constraint for the communication matrix, because such a path has an effect on the frame generation and the frame path.			
Base	ARObject,SignalPathConstraint			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
operation	SwcToSwcOperationArguments	*	aggr	Reference to the operation arguments of one operation which must not take the predefined way in the topology.
physicalChannel	PhysicalChannel	1..*	ref	The SwcToSwcSignal must not be transmitted on one of these physical channels.
signal	SwcToSwcSignal	*	aggr	The data element which must not take the predefined way in the topology.

Table 5.39: ForbiddenSignalPath

5.2.2.3 PermissibleSignalPath

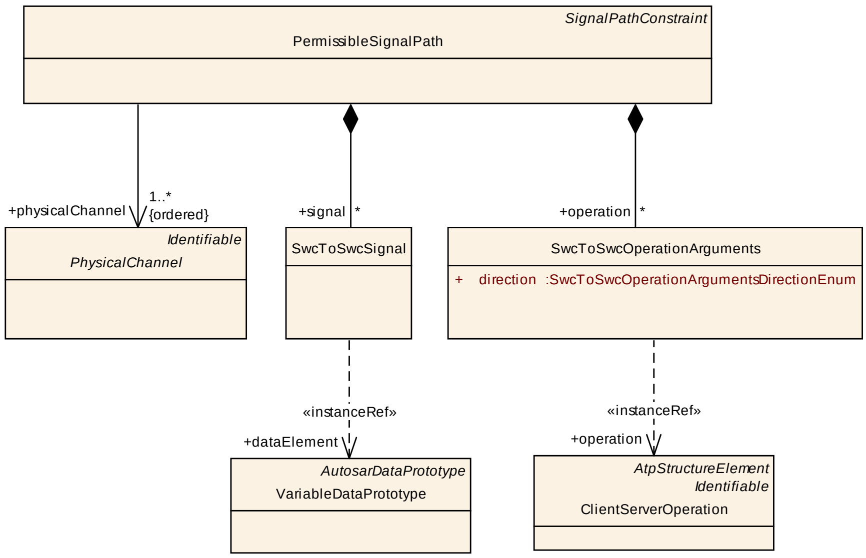

Figure 5.19: Description of the signal path that a signal must take in the topology (PermissibleSignalPath)

Class	**PermissibleSignalPath**			
Package	M2::AUTOSARTemplates::SystemTemplate::SignalPaths			
Note	The PermissibleSignalPath describes the way a data element shall take in the topology. The path is described by ordered references to PhysicalChannels. If more than one PermissibleSignalPath is defined for the same signal/operation attributes, any of them can be chosen. Such a signal path can be a constraint for the communication matrix . This path describes that one data element should take path A (e.g. 1. CAN channel, 2. LIN channel) and not path B (1. CAN channel, FlexRay channel A). This has an effect on the frame generation and the frame path.			
Base	ARObject,SignalPathConstraint			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
operation	SwcToSwcOperationArguments	*	aggr	The arguments of an operation that can take the predefined way in the topology.
physicalChannel (ordered)	PhysicalChannel	1..*	ref	The SwcToSwcSignal can be transmitted on one of these physical channels.
signal	SwcToSwcSignal	*	aggr	The data element which can take the predefined way in the topology.

Table 5.40: PermissibleSignalPath

5.2.2.4 SeparateSignalPath

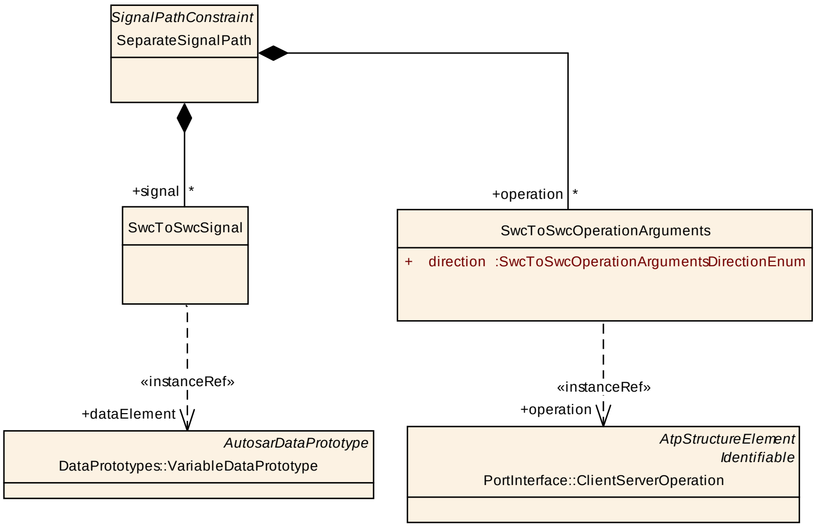

Figure 5.20: Description of signals that must not take the same way in the topology (SeparateSignalPath)

Class	**SeparateSignalPath**			
Package	M2::AUTOSARTemplates::SystemTemplate::SignalPaths			
Note	The SeparateSignalPath describes that two SwcToSwcSignals and/or SwcToSwcOperationArguments must not take the same way (Signal Path) in the topology (e.g. Redundancy). This means that the signals are not allowed to share even a single physical channel in their path.			
Base	ARObject,SignalPathConstraint			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
operation	SwcToSwcOperationArguments	*	aggr	The SwcToSwcOperationArguments that must not take the same way (Signal Path) in the topology.
signal	SwcToSwcSignal	*	aggr	The SwcToSwcSignals that must not take the same way (Signal Path) in the topology.

Table 5.41: SeparateSignalPath

5.3 RTE and basic software resource estimations

Important constraints for system partitioning are the available resources on the ECUs in the system. For SW components, the resource estimations can be stated in SW component descriptions. It is however not only SW components that require resources. AUTOSAR RTE and basic software running on the ECU have resource needs as well.

The realization of the RTE and the kind of basic software to be run on a certain ECU depend on the implicit and explicit usage of all basic software by the software components. The software components need to communicate internally and with software components on other ECUs. Furthermore, they have different needs with respect to scheduling. This results in implicit use of e.g. communication and operating system software. In addition, the software components make explicit use of basic software when they e.g. utilize system services (e.g. diagnostics) and access sensors/actuators via the I/O abstraction layer or the Complex Driver abstraction layer. Thus, the resource consumption of the RTE and the basic software depend on the SW Components mapped to the ECU, since this determines the exact configuration of the RTE and the basic software.

[TPS_SYST_01126] Resource Consumption for RTE and basic software ⌈ The resource consumption for RTE and basic software may be specified using class EcuResourceEstimation. Each estimation is performed for a specific ECU and for a specific set of SW mapped to that ECU (reference from EcuResourceEstimation to EcuInstance and SwcToEcuMapping). ⌋*(RS_SYST_00002)*

Different resource estimations for a specific ECU, but with different mappings may exist, e.g. for different variants of the system, or to show the difference of resource needs for different mappings. The EcuResourceEstimation aggregates the meta-class ResourceConsumption from the GenericStructure package each for RTE and basic software, which specifies stack and heap usage and execution time.

ExecutionTime and StackUsage are used to provide information on the implementation specific resource usage of the ExecutableEntity defined in the InternalBehavior of SW-Component respectively in the BswInternalBehavior of BSW Module. MemorySection documents the resources needed to load the object file containing the implementation on the ECU. HeapUsage describes the dynamic memory usage of the software.

Figure 5.21 shows the meta-model for resource estimations for RTE and basic SW.

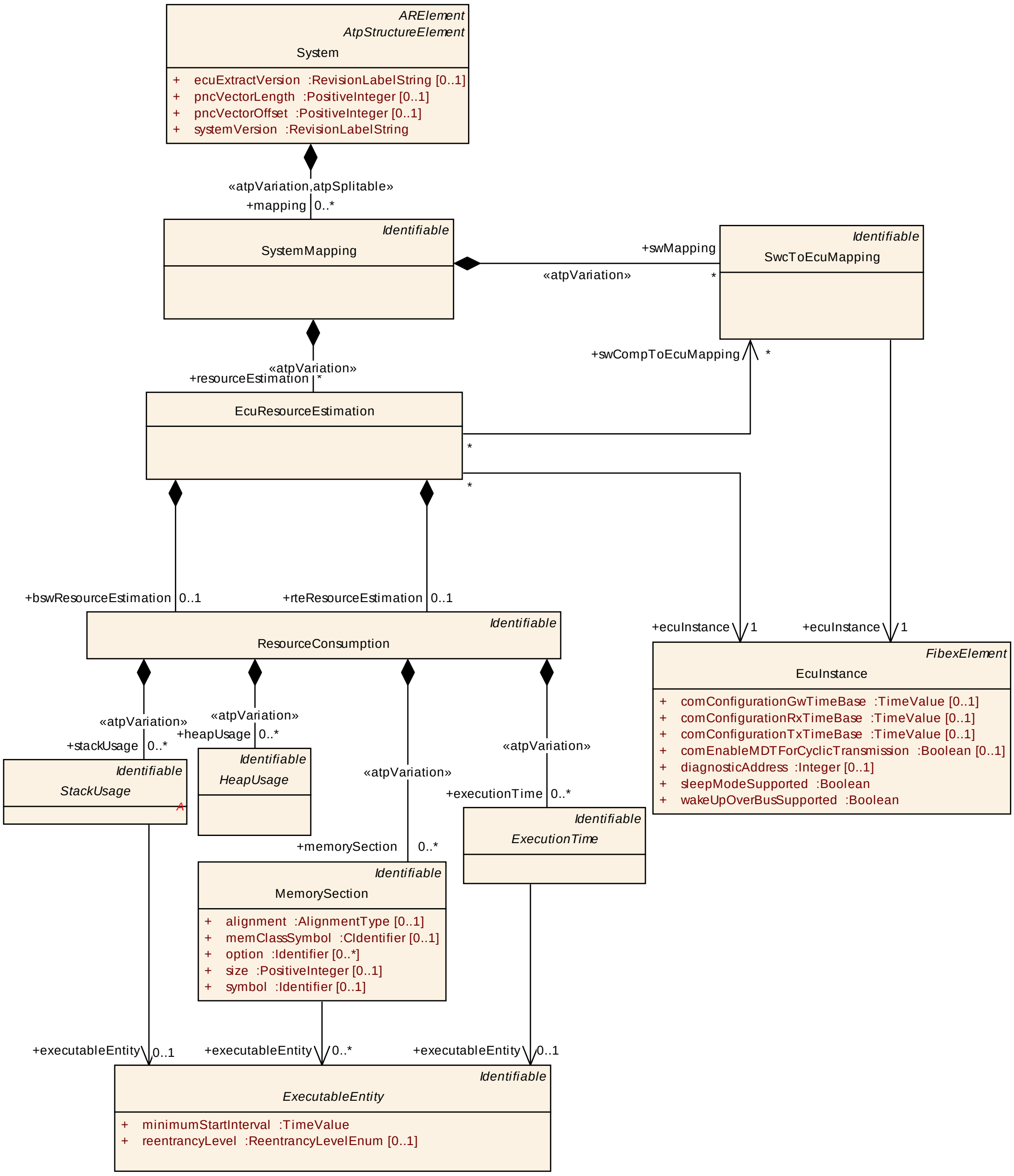

Figure 5.21: ECU resource estimations (ResourceEstimation)

[constr_3005] valid EcuResourceEstimation ⌈The same EcuInstance shall be referenced directly from the EcuResourceEstimation and from the SwcToEcuMapping:

EcuResourceEstimation.swCompToEcuMapping.ecuInstance == EcuResourceEstimation.ecuInstance ⌋

Class	**EcuResourceEstimation**			
Package	M2::AUTOSARTemplates::SystemTemplate::SWmapping			
Note	Resource estimations for RTE and BSW of a single ECU instance.			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
bswResourceEstimation	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
rteResourceEstimation	ResourceConsumption	0..1	aggr	Estimation for the resource consumption of the run time environment.
swCompToEcuMapping	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 5.42: EcuResourceEstimation

Class	**ResourceConsumption**			
Package	M2::AUTOSARTemplates::CommonStructure::ResourceConsumption			
Note	Description of consumed resources by one implementation of a software.			
Base	ARObject,Identifiable,MultilanguageReferrable,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
executionTime	ExecutionTime	*	aggr	Collection of the execution time descriptions for this implementation. The aggregation of executionTime is subject to variability with the purpose to support the conditional existence of runnable entities. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
heapUsage	HeapUsage	*	aggr	Collection of the heap memory allocated by this implementation. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
memorySection	MemorySection	*	aggr	An abstract memory section required by this Implementation. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Attribute	Datatype	Mul.	Kind	Note
sectionNamePrefix	SectionNamePrefix	*	aggr	A prefix to be used for the memory section symbol in the code. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
stackUsage	StackUsage	*	aggr	Collection of the stack memory usage for each runnable entity of this implementation. The aggregation of StackUsage is subject to variability with the purpose to support the conditional existence of runnable entities. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table 5.43: ResourceConsumption

The element ResourceConsumption and the subelements HeapUsage, MemorySection, StackUsage and ExecutionTime are described in more detail in the BSW Module Description [17].

5.4 Partial Networking

The AUTOSAR BSW stack supports power saving during vehicle operation time with the partial networking mechanism. This mechanism allows to shut down and startup the bus communication interfaces of groups of ECUs (Partial Network Cluster) during normal bus communication.

On the VFB Level Partial Networks are represented by Virtual Function Clusters and are described with PortGroups. The Virtual Function Cluster groups the communication necessary to realize one or more vehicle functions that can become activated/deactivated during normal vehicle operation. Virtual Function Clusters are described in more detail in [5]. The Virtual Function Clusters are mapped onto Partial Network Clusters.

[TPS_SYST_01133] Partial Network Clusters ⌈ Partial Network Clusters are realized with ISignalIPduGroups using PncMapping. ⌋*(RS_SYST_00042)*

PncMapping is Describable.

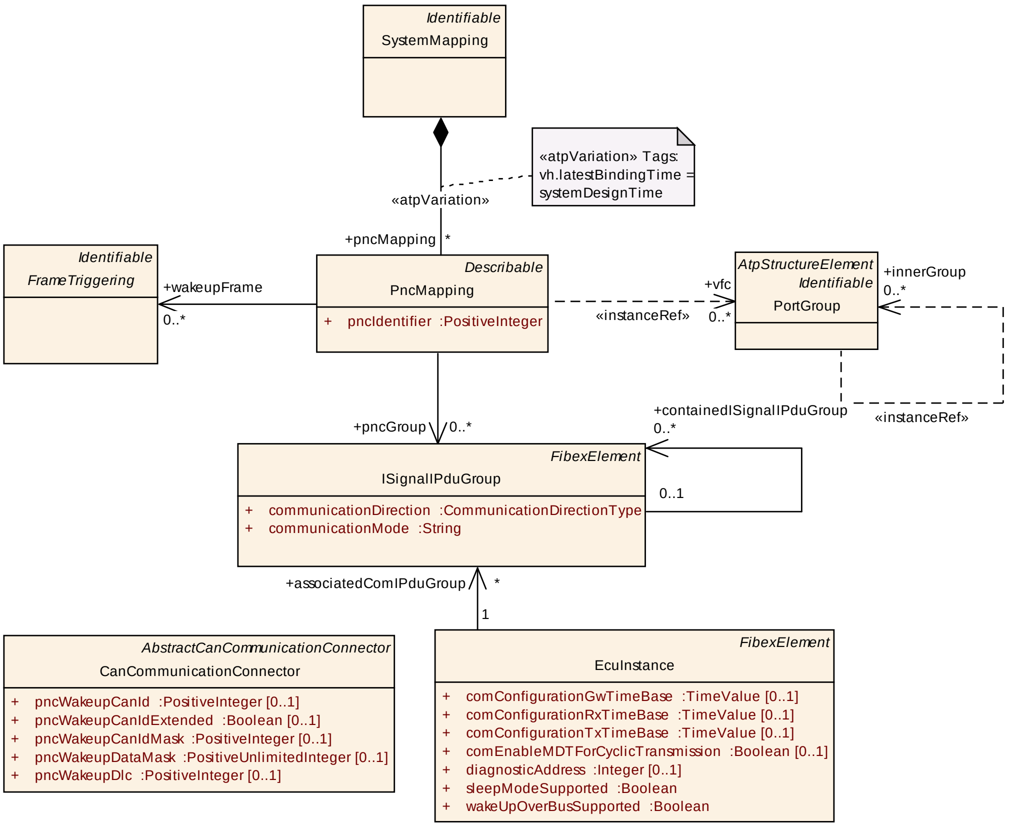

Figure 5.22: Mapping of Virtual Function Clusters onto Partial Network Clusters

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	***Datatype***	***Mul.***	***Kind***	***Note***
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.
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.

Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
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 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.

Table 5.44: PncMapping

[constr_3039] `pncIdentifier` range ⌈ The `pncIdentifier` value shall be in the range of 8..63. ⌋

The runtime information that is used to coordinate the request/release information of all partial networks is called `pncVector`. The size and position of the `pncVector` inside the network management user data (`NmPdu.iSignalToIPduMapping`) is globally defined in the `System` class in chapter 1.6.

In the system description the `NmPdu`s are described based on the actual network interaction (i.e. an ECU sends one `NmPdu` per network and receives a set of `NmPdu`s). Those `NmPdu` with user data have, according to section 6.1, `PduTriggering`s and assignments to `IPduPort`s defined.

The `NmPdu`s contributing to partial networking also have the Nm user data layout specified to contain the `pncVector` according to [constr_3043].

Those `Pdu`s which are used to perform the ECU internal communication between the basic software modules (like *EIRA*, *ERA*, *IRA*) are not described in the system description and need to be introduced to the ECU Configuration.

[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..6. ⌋

[constr_3043] `pncVector` configuration in AUTOSAR Com ⌈ The `pncVector` shall be configured as UINT8_N signal in AUTOSAR Com. ⌋

Attributes used to configure the Partial Network Wakeup of one specific Ecu are described in chapter 3.3.1.4.

6 Communication

This chapter describes all topics that deal with constraints or configurations that describe the information exchange between the ECUs. The description of communication matrices in the System Template is based on the description in ASAM FIBEX [9]. Because of the requirements of AUTOSAR some extensions were made to the original FIBEX model.

The main elements to describe communication in the System Template are SystemSignals, ISignals, Pdus and Frames, as it can be seen on Figure 6.1.

Frames can be defined independently of communication clusters. On the communication channel the Frame is represented by the referencing FrameTriggering.

A Frame has a payload section of a certain length in bytes, which contains an arbitrary number of non-overlapping Pdus. In AUTOSAR only FlexRay supports the packing and unpacking of multiple Pdus into/out of one FlexRay Frame. The AUTOSAR CanIf and LinIf are not capable of packing multiple Pdus into one Frame.

[constr_3036] Pdus in CAN and LIN Frames ⌈ CAN Frames and LIN Frames shall only contain one Pdu. ⌋

A Pdu (Protocol Data Unit) is the information delivered through a network layer. For the network to understand which layer is being discussed, a single-letter prefix is added to the PDU.

- IPdu - Interaction Layer Protocol Data Unit (assembled and disassembled in Com). In the case of external communication the Interaction Layer packs one or more signals into assigned IPdus and passes them to the underlying layer for transfer between nodes in a network.
- NPdu - Network Layer Protocol Data Unit (assembled and disassembled in a Transport Protocol module). The TP module's main purpose is the segmentation and reassembly of IPdus that do not fit in one of the assigned NPdus.
- LPdu - Data Link Layer Protocol Data Unit (assembled and disassembled in AUTOSAR Hardware Abstraction layer). The element Frame in the System Template represents the Autosar Layered Architectures LSdu. Sdu is the abbreviation of "Service Data Unit". The Data Link Layers LPdu contains the LSdu and PCI (Protocol Control Information). The LPdu is not described in the System Template.

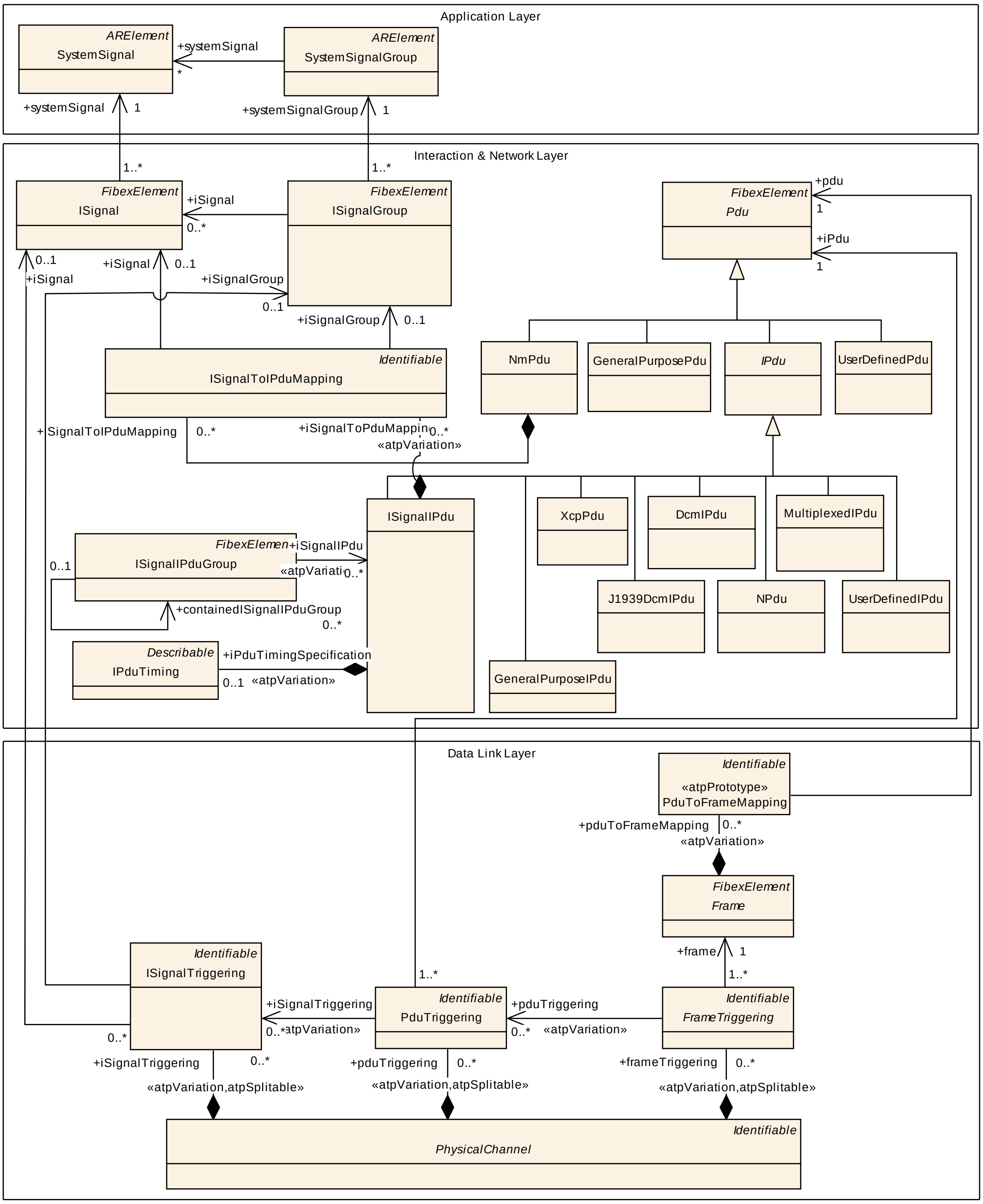

Figure 6.1: Communication Overview (FibexCore: Communication)

In case no multiplexing is performed the IPdus of Com that fit into one frame can be passed directly via the PDU Router to the communication interfaces.

[constr_3037] maximum Frame frameLength for CAN and LIN ⌈ For CAN and LIN the maximum frameLength is 8 bytes. ⌋

[constr_3038] maximum Frame frameLength for FlexRay ⌈ For FlexRay the maximum frameLength is 254 bytes. ⌋

[TPS_SYST_01048] Handling of large IPdus ⌈ Large IPdus that are too long to fit into one Frame of the respective subclass of CommunicationCluster shall be routed via a Transport Protocol to the communication interfaces. ⌋

For example an IPdu with the length of 10 bytes needs to be routed via a Transport Protocol on CAN but on FlexRay this is not required.

[TPS_SYST_01049] Handling of IPdus with dynamic signals ⌈ IPdus which contain dynamic signals shall be routed via a Transport Protocol to the communication interfaces. ⌋*(RS_SYST_00029, RS_SYST_00030)*

[TPS_SYST_01051] Handling of DcmIPdus ⌈ All DcmIPdus shall be transported via a Transport Protocol [1]. ⌋

The Transport Protocols are described in more detail in chapter 6.8.

If multiplexing is performed an IPdu is routed between the IPdu Multiplexer and the Interface Layer or Transport Layer. To distinguish these two different cases two specializations ISignalIPdu and MultiplexedIPdu are introduced. A ISignalIPdu represents an IPdu handled by AUTOSAR Com. The AUTOSAR IPduM is responsible to combine Com ISignalIPdus to MultiplexedIPdus. On receiver-side the IPduM is responsible to interpret the content of MultiplexedIPdus and provide Com separated ISignalIPdus by taking into account the value of the selector field. The IPdu Multiplexer is described in more detail in chapter 6.5.

AUTOSAR Com provides the possibility to define Transmission Modes for each Com ISignalIPdu. For this reason the ISignalIPdu aggregates the IPduTiming. The Transmission Modes are described in more detail in chapter 6.4.

6.1 Triggerings and Ports

The elements FrameTriggering, PduTriggering and ISignalTriggering are describing the usage of Frames, IPdus and ISignals on a PhysicalChannel.

A FrameTriggering need to fulfill requirements for contained Pdus that are defined by the corresponding PduTriggerings. And the PduTriggering need to fulfill requirements for contained ISignals that are defined by the corresponding ISignalTriggerings. The references between the Triggering elements can be used to describe these relationships. More details can be found in class tables of FrameTriggering, PduTriggering and ISignalTriggering.

In AUTOSAR the timing of bus messages can be controlled by send requests of the Application layer in combination with the Com Transmission Modes and Transfer Properties (esp. CAN). On the other hand it can be controlled by the FlexRay or LIN Interface. In this case the Bus Interface only requests IPdus that have to be provided by Com.

[1] There is one special gateway use case where a Transport Protocol NPdu can be routed directly by the Pdu Router and where the TP module is not involved. More details can be found in chapter 6.8.

In the System Template the Com controlled timing is described with the aggregation between the ISignalIPdu and the IPduTiming. The LIN and FlexRay Scheduling Tables are described in the FrameTriggering.

Timing requirements for FlexRay, TTCAN and LIN Pdus can be specified with the Timing Extension model. More details are described in chapter 1.7.3.

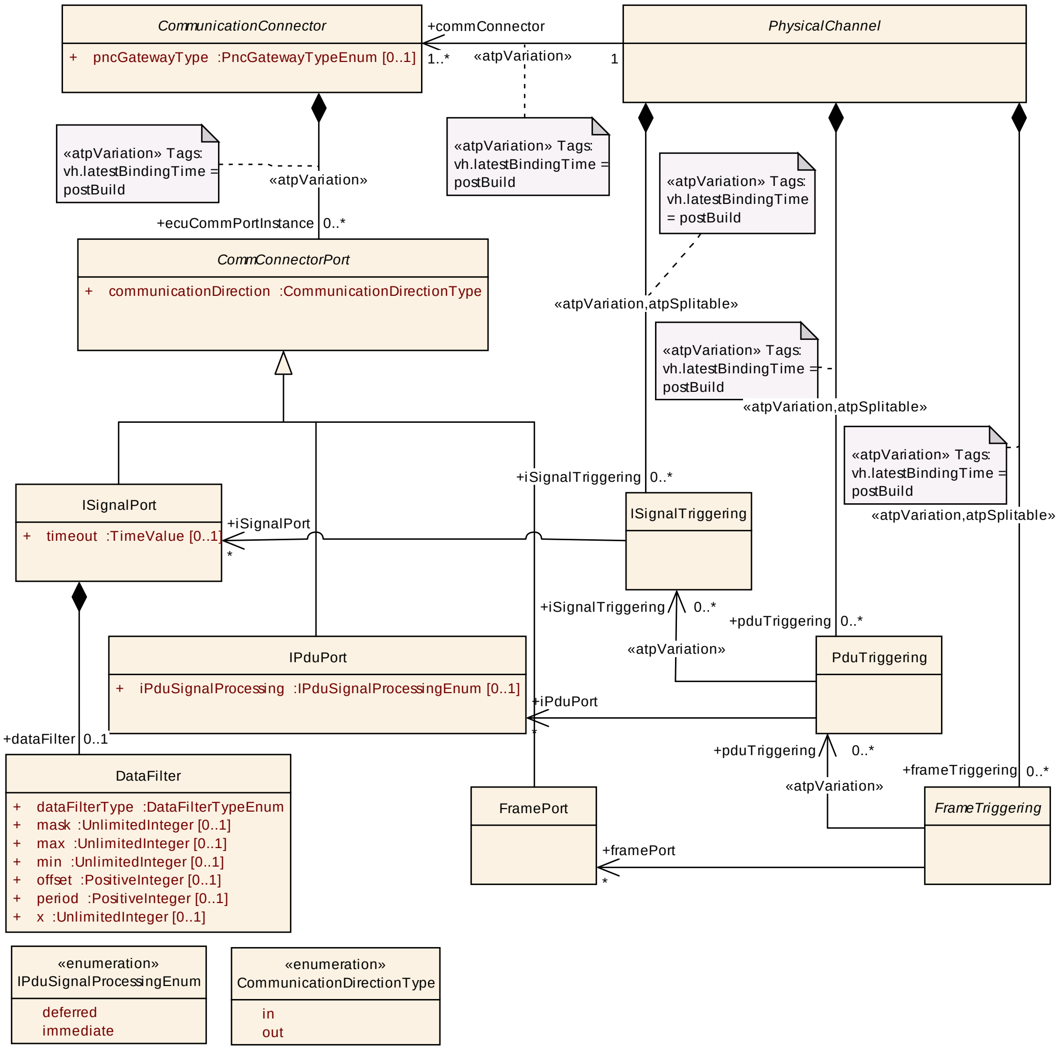

Figure 6.2: Communication Matrix (FibexCore: CommunicationMatrix)

Figure 6.2 shows the relationship between the CommConnectorPort and the FrameTriggering, PduTriggering and ISignalTriggering. This relationship allows to specify explicitly which Frames, Pdus, ISignals are received/sent by the connected ECU on the connected channel.

[TPS_SYST_01142] Rules for the creation of Triggerings and Ports on the sender side ⌈

- Application sends Signal
  - ISignalTriggering and ISignalPort shall be created

  - optional DataMapping between SystemSignal and VariableDataPrototype shall be created (in case of legacy signals this data mapping is not available).

- COM Signal Gateway
  - ISignalTriggering and ISignalPort shall be created.
  - ISignalMapping between two ISignalTriggerings shall be created (see chapter 7.3 for more details).

- Signal part of a Pdu but NOT sent by Application or Signal Gateway
  - Pdu part of Pdu Gateway
    * ISignalTriggerings for all contained signals shall be created (without connections to ISignalPort).
    * IPduMapping between two PduTriggerings shall be created (see chapter 7.2 for more details).
  - Pdu sent by Com
    * ISignalTriggerings for all contained signals shall be created (without connections to ISignalPort).

- Neither ISignal, Pdu nor Frame sent by the ECU
  - ISignalTriggering, PduTriggering and FrameTriggering on the PhysicalChannel shall be created. The connections to ISignalPort, IPduPort, FramePort shall be skipped.

⌋

The following rules apply for the creation of PduTriggerings and IPduPorts:

- **[TPS_SYST_01052] Routing of UserDefinedPdus, NmPdus, NPdus, GeneralPurposePdus** ⌈ UserDefinedPdus, NmPdus, NPdus, GeneralPurposePdus which are not going through the Pdu Router get their triggering information via the containing FrameTriggering and FramePort (no PduTriggering is defined for these Pdus). ⌋

- **[TPS_SYST_01053] Low-level routing of NPdus** ⌈ In case of a low-level routing of NPdus the Pdus are handled like IPdus and the PduTriggering and IPduPort shall be defined. ⌋

- **[TPS_SYST_01138] Low-level routing of XcpPdus** ⌈ Low-level routing of GeneralPurposeIPdus with category Xcp (or XcpPdus) : In case of a low-level routing of GeneralPurposeIPdus with category Xcp (or XcpPdus) the Pdus are handled like IPdus and the PduTriggering and IPduPort shall be defined. ⌋

- **[TPS_SYST_02005] Low-level routing of J1939DcmIPdus** ⌈ Low-level routing of GeneralPurposeIPdus with category J1939Dcm (or J1939DcmIPdu)

: In case of a low-level routing of GeneralPurposeIPdus with category J1939Dcm (or J1939DcmIPdus) the Pdus are handled like IPdus and the PduTriggering and IPduPort shall be defined. ⌋

- **[TPS_SYST_01054] Routing of DcmIPdus** ⌈ DcmIPdus shall have PduTriggering and IPduPorts since they are handled by the PduR (connection to the Dcm and/or DcmIPdu-routing). ⌋
- **[TPS_SYST_01055] Routing of ISignalIPdus that are part of a MultiplexedIPdu** ⌈ ISignalIPdus that are part of a MultiplexedIPdu (static or dynamic) and are also handled by the Com module shall have a PduTriggering and IPduPorts since they are handled by the PduR (and Com). Especially it is allowed to ignore certain received parts of a MultiplexedIPdu in a specific ECU. ⌋
- **[TPS_SYST_01056] Routing of ISignalIPdus, UserDefinedIPdus, MultiplexedIPdus, GeneralPurposeIPdus** ⌈ ISignalIPdus (not part of MultiplexedIPdus), UserDefinedIPdus, MultiplexedIPdus and GeneralPurposeIPdus shall have a PduTriggering and IPduPort if they are handled by the PduR. Especially it is allowed to ignore a certain IPdu out of a Flexray frame if it is not considered in a specific ECU. ⌋
- **[TPS_SYST_01057] Routing of NmPdus** ⌈ In case a NmPdu contains user data and is handled by the BusNm via the PduR and Com the NmPdu gets PduTriggering and IPduPort. ⌋

The following rule applies to the creation of ISignalTriggering and ISignalPort:

[TPS_SYST_01058] Pdu Gateway where an Ecu only routes a PduTriggering without being interested in the content ⌈ In case of a Pdu Gateway where an Ecu only routes a PduTriggering without being interested in the content, the reference between the ISignalTriggerings (that are referred to by the PduTriggering in the role iSignalTriggering) and the respective ISignalPorts shall not be created. ⌋

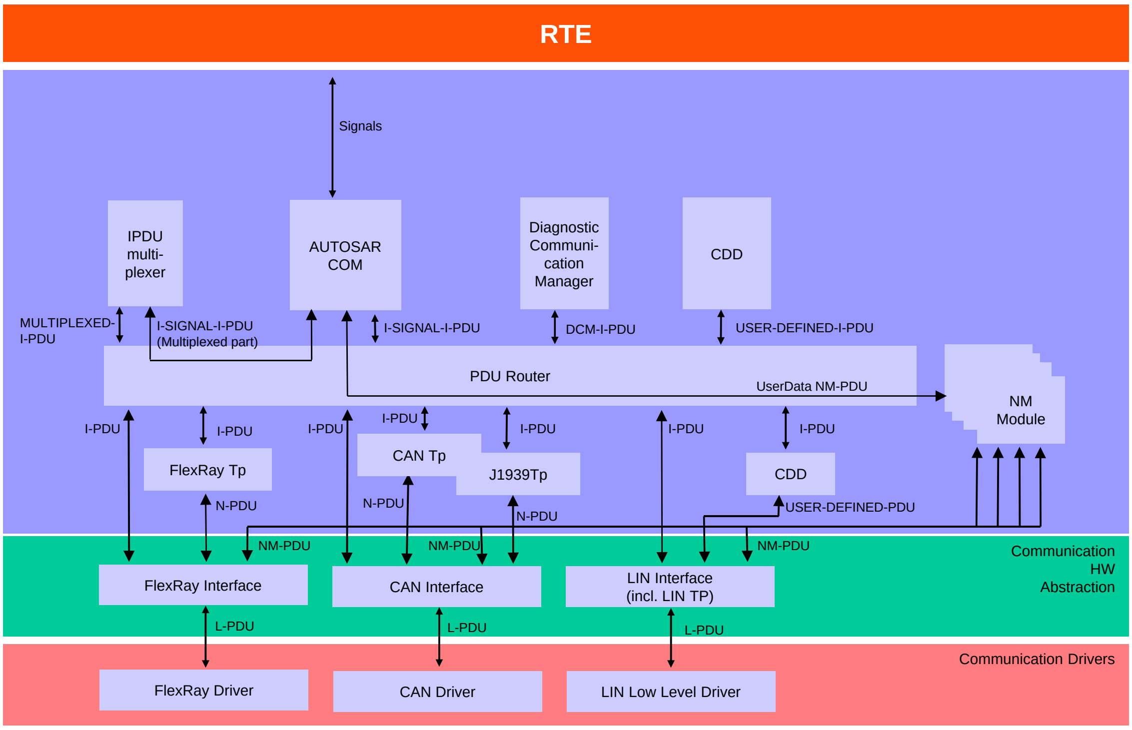

Figure 6.3: AUTOSAR Layered Architecture

Class	**CommConnectorPort (abstract)**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology			
Note	The Ecu communication relationship defines which signals, Pdus and frames are actually received and transmitted by this ECU. For each signal, Pdu or Frame that is transmitted or received and used by the Ecu an association between an ISignalPort, IPduPort or FramePort with the corresponding Triggering shall be created. An ISignalPort shall be created only if the corresponding signal is handled by COM (RTE or Signal Gateway). If a Pdu Gateway ECU only routes the Pdu without being interested in the content only a FramePort and an IPduPort needs to be created.			
Base	ARObject,Identifiable,MultilanguageReferrable,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
communicationDirection	CommunicationDirectionType	1	attr	Communication Direction of the Connector Port (input or output Port).

Table 6.1: CommConnectorPort

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	***Datatype***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Attribute	*Datatype*	*Mul.*	*Kind*	*Note*

Table 6.2: FramePort

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	*Datatype*	*Mul.*	*Kind*	*Note*
iPduSignalProcessing	IPduSignalProcessingEnum	0..1	attr	Definition of the two signal processing modes Immediate and Deferred for both Tx and Rx IPdus.

Table 6.3: IPduPort

Enumeration	**IPduSignalProcessingEnum**
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication
Note	Definition of signal processing modes.
Literal	*Description*
deferred	The signal indications / confirmations are deferred.
immediate	The signal indications / confirmations are performed.

Table 6.4: IPduSignalProcessingEnum

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	*Datatype*	*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.

Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
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 ReceiverComSpec, in this case the timeout value in ReceiverComSpec override this optional timeout specification.

Table 6.5: ISignalPort

[TPS_SYST_01059] Relationship between FrameTriggering and CommConnectorPort ⌈ For the reference between FrameTriggering and FramePort two approaches are supported:

- One to One relationship between FrameTriggering and FramePort per EcuInstance
- One FramePort per communicationDirection per EcuInstance exists and is referenced by all applicable FrameTriggerings (n to 1).

⌋

[TPS_SYST_01060] Relationship between PduTriggering and CommConnectorPort ⌈ For the reference between PduTriggering and IPduPort two approaches are supported:

- One to One relationship between PduTriggering and IPduPort per EcuInstance
- One IPduPort per communicationDirection per EcuInstance exists and is referenced by all applicable PduTriggerings (n to 1).

⌋

[TPS_SYST_01061] Relationship between ISignalTriggering and CommConnectorPort ⌈ For the reference between ISignalTriggering and ISignalPort two approaches are supported:

- One to One relationship between ISignalTriggering and ISignalPort per EcuInstance
- One ISignalPort per communicationDirection per timeout per EcuInstance exists and is referenced by all applicable PduTriggerings (n to 1).

⌋

6.2 ISignals

SystemSignals can be defined independently of CommunicationClusters and are representing the VariableDataPrototypes, ArgumentDataPrototypes, Triggers and ModeDeclarationGroupPrototypes in the communication description.

The RTE supports a "signal fan-out" where the same signal (System Signal) is sent in different IPdus to multiple receivers. The Pdu Router supports the "PDU fan-out" where the same IPdu is sent to multiple destinations.

To support the "signal fan-out" ISignals and ISignalGroups are introduced. An ISignal(ISignalGroup) represents the SystemSignal(SystemSignalGroup) of the Interaction Layer.

In case of "signal fan-out", several ISignals in different IPdus refer to the same SystemSignal. The "Signal fan-out" will be executed by the RTE. ISignals describe the Interface between the precompile configured RTE and the potentially postbuild configured Com Stack.

The ISignalToIPduMapping element describes the mapping of ISignals to ISignalIPdus and defines the position of an ISignal within an ISignalIPdu.

[constr_3009] Overlapping of ISignals is prohibited ⌈ ISignals mapped to an ISignalIPdu shall not overlap. ⌋

[constr_3010] ISignalIPdu length shall not be exceeded ⌈ The combined length of all ISignals and updateIndicationBits that are mapped into an ISignalIPdu shall not exceed the defined Pdu length. ⌋

[constr_3011] Overlapping of updateIndicationBits of ISignals is prohibited ⌈ The updateIndicationBitPosition for an ISignal in an ISignalIPdu shall not overlap with other updateIndicationBitPositions or ISignal locations. ⌋

[TPS_SYST_01062] Network representation of an ISignal ⌈ With the aggregation of SwDataDefProps in the role networkRepresentationProps the actual representation of the ISignal on the network can be specified. ⌋*(RS_SYST_00047)*

[TPS_SYST_01063] Context of network representation of an ISignal ⌈ The dataTypePolicy defines from which context the network representation specification shall be taken. ⌋*(RS_SYST_00001, RS_SYST_00047)*

For an alternative network representation it is important to define an alternative SwDataDefProps especially SwBaseType defining alternative encoding (e.g. from float in PortInterface to integer on bus).

[constr_3060] Usage of networkRepresentationProps and physicalProps ⌈ Usage of networkRepresentationProps and physicalProps shall follow the restrictions given in table 6.6. ⌋

	Element	
Attributes of SwDataDefProps	**SystemSignal.physicalProps**	**ISignal.networkProps**
additionalNativeTypeQualifier	NA	NA
annotation	NA	NA
baseType	NA	D
compuMethod	D	I
dataConstr	D	M
displayFormat	D	M
implementationDataType	NA	NA
invalidValue	NA	D
mcFunction	NA	NA
swAddrMethod	NA	NA
swAlignment	NA	NA
swBitRepresentation	NA	NA
swCalibrationAccess	NA	NA
swCalprmAxisSet	NA	NA
swCalprmAxisSet. swCalprmAxis /SwAxisGrouped. swCalprmRef	NA	NA
swCalprmAxisSet. swCalprmAxis /SwAxisIndividual. swVariableRef	NA	NA
swCalprmAxisSet. swCalprmAxis /SwAxisGrouped. sharedAxisType	NA	NA
swCalprmAxisSet. swCalprmAxis /SwAxisIndividual. inputVariableType	NA	NA
swCalprmAxisSet/ AxisIndividual/ 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
swRecordLayout	NA	NA
swRefreshTiming	NA	NA
swTextProps	NA	NA
swValueBlockSize	NA	NA
unit	D	M
valueAxisDataType	NA	NA

Table 6.6: Allowed SwDataDefProps Attributes for the ISignal and SystemSignal

The following settings apply in table 6.6:

D Define the attribute independent from settings to the left.

I Inherit the definition from the left for usage in the scope of this element.

NA Attribute is **not applicable** for usage in the scope of this element.

M Attribute is **meaningless** in the scope of this element. As it was allowed in previous versions, declaring it as Not Applicable (NA) would break compatibility. Tools shall ignore such an attribute without a warning.

[constr_3061] CompuMethod specification in networkRepresentationProps ⌈ A CompuMethod that is defined in the networkRepresentationProps for the ISignal shall be compatible to the CompuMethod that is defined in the physicalProps for the SystemSignal that is referenced by the ISignal. ⌋

In case that the System Description doesn't use a complete Software Component Description (VFB View) the physicalProps and networkRepresentationProps are used to configure the Data Semantics.

The networkRepresentationProps contains a reference to the SwBaseType. This reference can be used for the derivation of the ComSignalType in the AUTOSAR Com Configuration.

[TPS_SYST_02001] networkRepresentationProps are mandatory in case the dataTypePolicy is set to override or legacy ⌈ If the dataTypePolicy of an ISignal is set to override or legacy, the networkRepresentationProps for the respective ISignal have to be specified. ⌋

[TPS_SYST_02006] Usage of networkRepresentationFromComSpec ⌈If the networkRepresentationFromComSpec is used either the SwDataDefProps in the role networkRepresentation aggregated by the SenderComSpec or ReceiverComSpec shall exist or the ImplementationDataType shall exist. ⌋

[TPS_SYST_01141] Derivation of ComSignalType ⌈ The ComSignalType shall be derived from the baseTypeSize (or maxBaseTypeSize) and the baseTypeEncoding. ⌋

[TPS_SYST_01065] Mapping onto the of ComSignalType enumeration ⌈ The following table shows how the mapping onto the ComSignalType enumeration is done:

BaseTypeEncoding	*BaseTypeSize*	*ComSignalType*
2C	8 bits	SINT8
2C	16 bits	SINT16
2C	32 bits	SINT32
NONE	8 bits	UINT8
NONE	16 bits	UINT16
NONE	32 bits	UINT32
IEE754	32 bits	FLOAT32
IEE754	64 bits	FLOAT64
ISO-8859-1	baseTypeSize	UINT8_N (from the definition of baseTypeSize the ComSignalLength can be determined)

BaseTypeEncoding	*BaseTypeSize*	*ComSignalType*
ISO-8859-2	baseTypeSize	UINT8_N (from the definition of baseTypeSize the ComSignalLength can be determined)
WINDOWS-1252	baseTypeSize	UINT8_N (from the definition of baseTypeSize the ComSignalLength can be determined)
UTF-8	baseTypeSize	UINT8_N (from the definition of baseTypeSize the ComSignalLength can be determined)
UCS-2	baseTypeSize	UINT8_N (from the definition of baseTypeSize the ComSignalLength can be determined)
-	maxBaseTypeSize	UINT8_DYN (from the definition of maxBaseTypeSize the ComSignalLength can be determined)
BOOLEAN	-	BOOLEAN

Table 6.7: SwBaseType to ComSignalType Mapping

⌋*(RS_SYST_00029)*

The invalidValue is aggregated by the SwDataDefProps element. The SwDataDefProps and the SwBaseType classes are described in more detail in the Software Component Template [5].

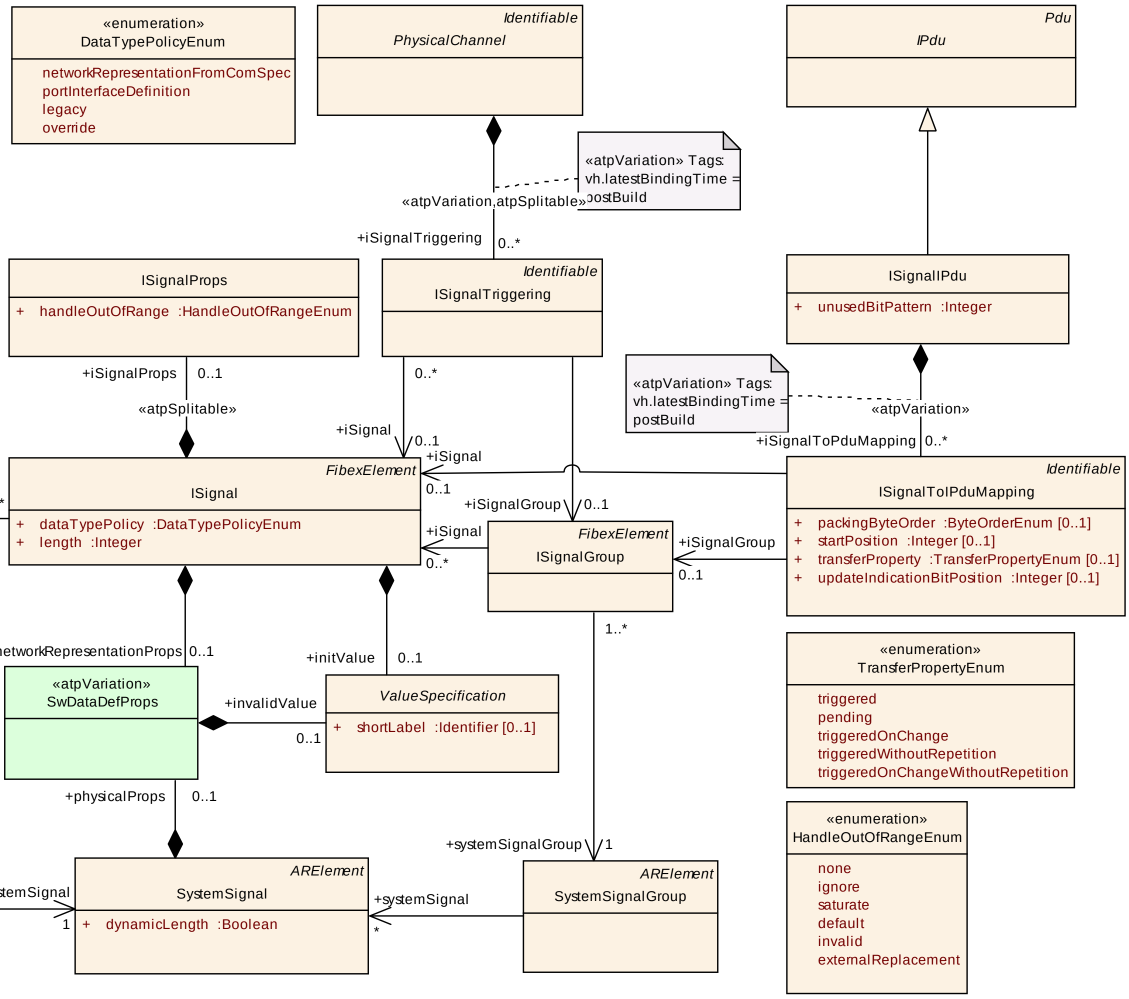

Figure 6.4: ISignals and the mapping into IPdus (FibexCore: SignalOverview)

The configuration of the COM Module for atomic signals can largely be derived from the System Template.

[TPS_SYST_01066] Derivation of Tx COM Signals ⌈ A COM signal shall be defined in the COM module configuration for each `ISignalToIPduMapping` that is aggregated by `ISignalIPdu` that in turn is referenced by an `PduTriggering` that in turn references an `IPduPort` where the `communicationDirection` is set to `out` of the regarded ECU. ⌋

[TPS_SYST_01067] Derivation of Rx COM Signals ⌈ A COM signal shall be defined in the COM module configuration for each `ISignalToIPduMapping` that is aggregated by `ISignalIPdu` that in turn is referenced by an `PduTriggering` that in turn references an `IPduPort` where the `communicationDirection` is set to `in` of the regarded ECU if the application requires access to the respective `SystemSignal`. ⌋

Please note that it is also possible to create a COM signal (even if the application does not require access to the respective `SystemSignal`) in case of a signal gateway.

To support the AUTOSAR concept of composite data types the AUTOSAR COM layer provides signal groups. Every record or array element of a composite data type requires a SystemSignal for the transmission. But the RTE has to guarantee the consistent transmission of data.

[TPS_SYST_01153] Atomic transport of SystemSignalGroups ⌈ A SystemSignalGroup shall be transmitted and received consistently; therefore it provides data consistency for composite data types. ⌋

A SystemSignalGroup refers to a set of SystemSignals that shall always be kept together in a common IPdu. An ISignalGroup represents a SystemSignalGroup of the Interaction Layer. In the case of "signal fan-out", several ISignalGroups refer to the same SystemSignalGroup.

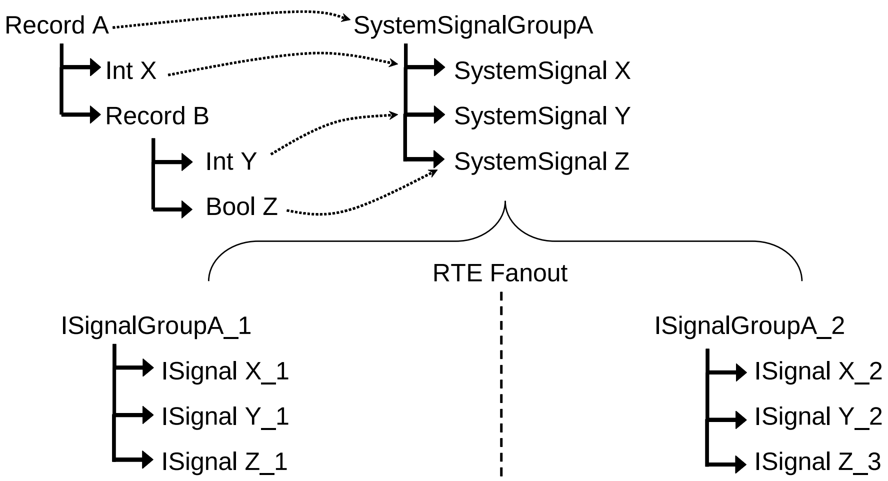

Figure 6.5: ISignal example

The example in Figure 6.5 shows the usage of ISignalGroups and ISignals. In this example a record is mapped to a SystemSignalGroup. All ApplicationRecordElements with ApplicationPrimitiveDataType are mapped to individual SystemSignals. If the same SystemSignalGroup is sent to different receivers (RTE Fanout) then two different ISignalGroups are created. For each SystemSignal within the SystemSignalGroup an ISignal is created. The different ISignals of the same SystemSignal can have different network representations.

[TPS_SYST_01156] Definition of ISignalTriggerings is allowed for ISignalGroups and for GroupSignals ⌈ If an ISignalGroup is referenced by an ISignalTriggering then the ISignals that are contained in the ISignalGroup (GroupSignals) may be referenced as well by ISignalTriggerings. ⌋

[constr_3094] Consistent ISignalPort.communicationDirection for ISignalTriggerings of ISignalGroups and contained ISignals ⌈ In case the ISignals 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. ⌋

[TPS_SYST_01157] Allowed usage of attributes for `ISignal`s, `ISignalGroup`s and `GroupSignal`s ⌈ The following table shows attributes that may be used to configure `ISignal`s that are not part of an `ISignalGroup`, `ISignalGroup`s and `ISignal`s that are part of an `ISignalGroup` (`GroupSignal`s):

	Element		
Attributes	**ISignal**	**ISignalGroup**	**GroupSignal**
`startPosition`	1	NA	1
`updateIndicationBitPosition`	0..1	0..1	NA
`transferProperty`	0..1	0..1	0..1
`packingByteOrder`	1	NA	1
`dataFilter`	0..1	NA	NA

Table 6.8: Allowed usage of attributes for `ISignal`s, `ISignalGroup`s and `GroupSignals`

⌋

[constr_3067] `initValue` defined in the context of `ISignal` ⌈ The definition of an `initValue` in the context of an `ISignal` can only be a primitive `NumericalValueSpecification` or `TextValueSpecification`. ⌋

[TPS_SYST_02012] `initValue` and `invalidValue` represent internal values ⌈ The `initValue` and `invalidValue` aggregated by the `networkRepresentationProps` shall represent the internal values. ⌋

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,Multilanguage Referrable,PackageableElement,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***

Attribute	Datatype	Mul.	Kind	Note
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
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 "InitValue". If a full DataMapping exist for the SystemSignal this information may be available from a configured SenderComSpec 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.

Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
networkRepresentationProps	SwDataDefProps	0..1	aggr	Specification of the actual network representation. The usage of SwDataDefProps for this purpose is restricted to the attributes compuMethod and baseType. The optional baseType attributes "memAllignment" and "byteOrder" shall not be used. The attribute "dataTypePolicy" in the SystemTemplate element defines whether this network representation shall be ignored and the information shall be taken over from the network representation of the ComSpec. If "override" is chosen by the system integrator the network representation can violate against the requirements defined in the PortInterface and in the network representation of the ComSpec. In case that the System Description doesn't use a complete Software Component Description (VFB View) this element is used to configure "ComSignalDataInvalidValue" and the Data Semantics.
systemSignal	SystemSignal	1	ref	Reference to the System Signal that is supposed to be transmitted in the ISignal.

Table 6.9: ISignal

Enumeration	**DataTypePolicyEnum**
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping
Note	This class lists the supported DataTypePolicies.
Literal	***Description***
legacy	In case the System Description doesn't use a complete Software Component Description (VFB View) this value can be chosen. This supports the inclusion of legacy signals. The aggregation of SwDataDefProps shall be used to configure the "ComSignalDataInvalidValue" and the Data Semantics.
networkRepresentationFromComSpec	Ignore any networkRepresentationProps of this ISignal and use the networkRepresentation from the ComSpec. Please note that the usage does not imply the existence of the SwDataDefProps in the role networkRepresentation aggregated by the SenderComSpec or ReceiverComSpec if an ImplementationDataType is defined.
override	If this value is chosen the requirements specified in the ComSpec (networkRepresentationFromComSpec) are not fullfilled by the aggregated SwDataDefProps. In this case the networkRepresentation is specified by the aggregated swDataDefProps.

portInterface Definition	This enumeration literal is deprecated and will be removed in future. Old description: Ignore any networkRepresentationProps of this ISignal and use the networkRepresentation specified in the VariableDataPrototypes owned by PortInterface (portInterfaceDefinition). **Tags:** atp.Status=obsolete

Table 6.10: DataTypePolicyEnum

Class	**ISignalProps**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Additional ISignal properties that may be stored in different files.			
Base	ARObject			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
handleOutOfRange	HandleOutOfRangeEnum	1	attr	This attribute defines the outOfRangeHandling for received and sent signals.

Table 6.11: ISignalProps

Enumeration	**HandleOutOfRangeEnum**
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication
Note	A value of this type is taken for controlling the range checking behavior of the AUTOSAR RTE.
Literal	***Description***
default	The RTE will use the initValue if the actual value is out of the specified bounds.
externalReplacement	This indicates that the value replacement is sourced from the externalReplacement.
ignore	The RTE will ignore any attempt to send or receive the corresponding dataElement if the value is out of the specified range.
invalid	The RTE will use the invalidValue if the value is out of the specified bounds.
none	A range check is not required.
saturate	The RTE will saturate the value of the dataElement such that it is limited to the applicable upper bound if it is greater than the upper bound. Consequently, it is limited to the applicable lower bound if the value is less than the lower bound.

Table 6.12: HandleOutOfRangeEnum

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,Multilanguage Referrable,PackageableElement,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
iSignal	ISignal	*	ref	Reference to a set of ISignals that shall always be kept together.
systemSignalGroup	SystemSignalGroup	1	ref	Reference to the SystemSignalGroup that is defined on VFB level and that is supposed to be transmitted in the ISignalGroup.

Table 6.13: ISignalGroup

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,Multilanguage Referrable,PackageableElement,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
systemSignal	SystemSignal	*	ref	Reference to a set of SystemSignals that must always be kept together.

Table 6.14: SystemSignalGroup

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	**Datatype**	**Mul.**	**Kind**	**Note**

Attribute	Datatype	Mul.	Kind	Note
iSignal	ISignal	0..1	ref	Reference to a ISignal that is mapped into the ISignalIPdu. 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.
packingByteOrder	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 packingByteOrder attribute). In AUTOSAR the bit counting is always set to "sawtooth" and the bit order is set to "Decreasing". 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.

Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
transferProperty	TransferPropertyEnum	0..1	attr	The triggered or triggeredOnChange, triggeredWithoutRepetition and triggeredOnChangeWithoutRepetition transferProperty causes immediate transmission of the IPdu, except if transmission mode Periodic or transmission mode NONE is defined for the IPdu. The Pending transfer property does not cause transmission of an I-PDU. The immediate transmission of the IPdu is caused even if only one Signal of an IPdu has the transferProperty triggered or triggeredWithoutRepetition or triggeredOnChange or triggeredOnChangeWithoutRepetition and all other Signals have the transferProperty pending. Also for ISignals of an ISignalGroup (GroupSignals) this attribute is relevant and shall be evaluated: • If none of the ISignals belonging to the ISignalGroup have a transferProperty defined the transferProperty of the ISignalToPduMapping referring to the ISignalGroup is considered. • If at least one of the ISignals belonging to the ISignalGroup has a transferProperty defined all other ISignals belonging to the same ISignalGroup shall have a transferProperty defined as well. All of the transferProperties of the GroupSignals are considered.
updateIndicationBitPosition	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 updateIndicationBitPosition is linked to the value of the attribute packingByteOrder because the method of finding the bit position is different for the values mostSignificantByteFirst and mostSignificantByteLast. This means that if the value of packingByteOrder is changed while the value of updateIndicationBitPosition remains unchanged the exact bit position of updateIndicationBitPosition within the enclosing ISignalIPdu still undergoes a change.

Table 6.15: ISignalToIPduMapping

[constr_3514] No two ISignalToIPduMappings shall reference the identical ISignal ⌈ No two ISignalToIPduMappings shall reference the identical ISignal in the role iSignal. ⌋

Enumeration	**TransferPropertyEnum**
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::Core Communication
Note	Transfer Properties of a Signal.
Literal	*Description*
pending	If the signal has the TransferProperty pending, then the function Com_SendSignal shall not perform a transmission of the IPdu associated with the signal.
triggered	The signal in the assigned IPdu is updated and a request for the IPdu's transmission is made.
triggeredOn Change	The signal in the assigned IPdu is updated and a request for the IPdus transmission is made only if the signal value is different from the already stored signal value.
triggeredOn ChangeWith-outRepetition	The signal in the assigned IPdu is updated and a request for the IPdus transmission is made only if the signal value is different from the already stored signal value. In the DIRECT/N-TIMES or MIXED transmission mode (EventControlledTiming) the IPdu will be transmitted just once without a repetition, independent of the defined NumberOfRepeats.
triggered Without Repetition	The signal in the assigned IPdu is updated and a request for the IPdu's transmission is made. In the DIRECT/N-TIMES or MIXED transmission mode (EventControlledTiming) the IPdu will be transmitted just once without a repetition, independent of the defined NumberOfRepeats.

Table 6.16: TransferPropertyEnum

[constr_3024] Usage of triggeredWithoutRepetition and triggeredOnChangeWithoutRepetition is not allowed for signal groups and group signals. ⌈ The values triggeredWithoutRepetition and triggeredOnChangeWithoutRepetition shall not be used if the ISignalToIPduMapping refers to an ISignalGroup or an ISignal which is part of an ISignalGroup (group signal). ⌋

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	*Datatype*	*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.
iSignalGro up	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 ISignalTriggering-ISignal reference.

Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
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 6.17: ISignalTriggering

6.2.1 Big Endian and Little Endian memory layout of Pdus and Frames

The AUTOSAR system description provide means to specify how the memory layout looks like when signals are packed into Pdus and Pdus are packed into Frames. The layout of Pdus and Frames on different communication systems is out of scope of AUTOSAR. The specification of attributes Bit counting (monotone or sawtooth) and Bit order (decreasing or increasing)[2] is not supported by AUTOSAR. In AUTOSAR these attributes are fixed.

[TPS_SYST_01068] Bit Counting in AUTOSAR ⌈ The Bit counting shall always be considered as "sawtooth". ⌋

[TPS_SYST_01069] Bit Order in AUTOSAR ⌈ The bit order shall always be considered as "Decreasing". ⌋

When a signal is mapped into a Pdu only the packingByteOrder affects the memory layout of the signal inside the Pdu beginning with it´s start bit position.

Little endian stores the least significant byte first and begins with the least significant bit, i.e. loworder bit in the sequence (the least significant bit serves as start bit).

Big endian stores the most significant byte first and begins with the most significant bit, i.e. the bit with the greatest numerical value (the most significant bit serves as start bit).

In both cases the bit positions in the mapped signals increase with the bit positions in the ISignalIPdu such that the bit 2^0 is mapped to position n in the ISignalIPdu and bit 2^1 is mapped to position n+1 and so on.

Example 6.6 shows the memory layout for Little Endian and Big Endian if an ISignal with a length of 10 bits is mappped into a Pdu. The start bit position is 5.

[2]More details about Bit counting and Bit order can be found in ASAM FIBEX [9].

Little Endian byte order:

Byte	0								1							
Bit	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8
Signal	2^2	2^1	2^0	-	-	-	-	-	-	2^9	2^8	2^7	2^6	2^5	2^4	2^3

Big Endian byte order:

Byte	0								1							
Bit	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8
Signal	-	-	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	-	-	-	-

Figure 6.6: PackingByteOrder Example

The mapping of `Pdu`s into `Frame`s is handled in the same way as the mapping of signals into `Pdu`s.

6.3 PDUs

The PDU Router deploys ISignalIPdus, DcmIPdus, UserDefinedIPdus, GeneralPurposeIPdus and MultiplexedIPdus onto different communication protocols. The PDU Router also determines if a transport protocol has to be used or not. Additional to the already mentioned Pdu Types the following types exist: NmPdu, NPdu, GeneralPurposePdus, UserDefinedPdus. Usually these Pdus are not routed by the PDU Router. [3]

UserDefinedPdus and UserDefinedIPdus are used to describe PDU-based communication over Complex Drivers. Chapter 6.11 provides a more detailed description of CDDs.

[3] There is one special gateway use case where a NPdu or a GeneralPurposeIPdu with the category XcpPdu is routed by the Pdu Router. More details can be found in chapter 6.8.

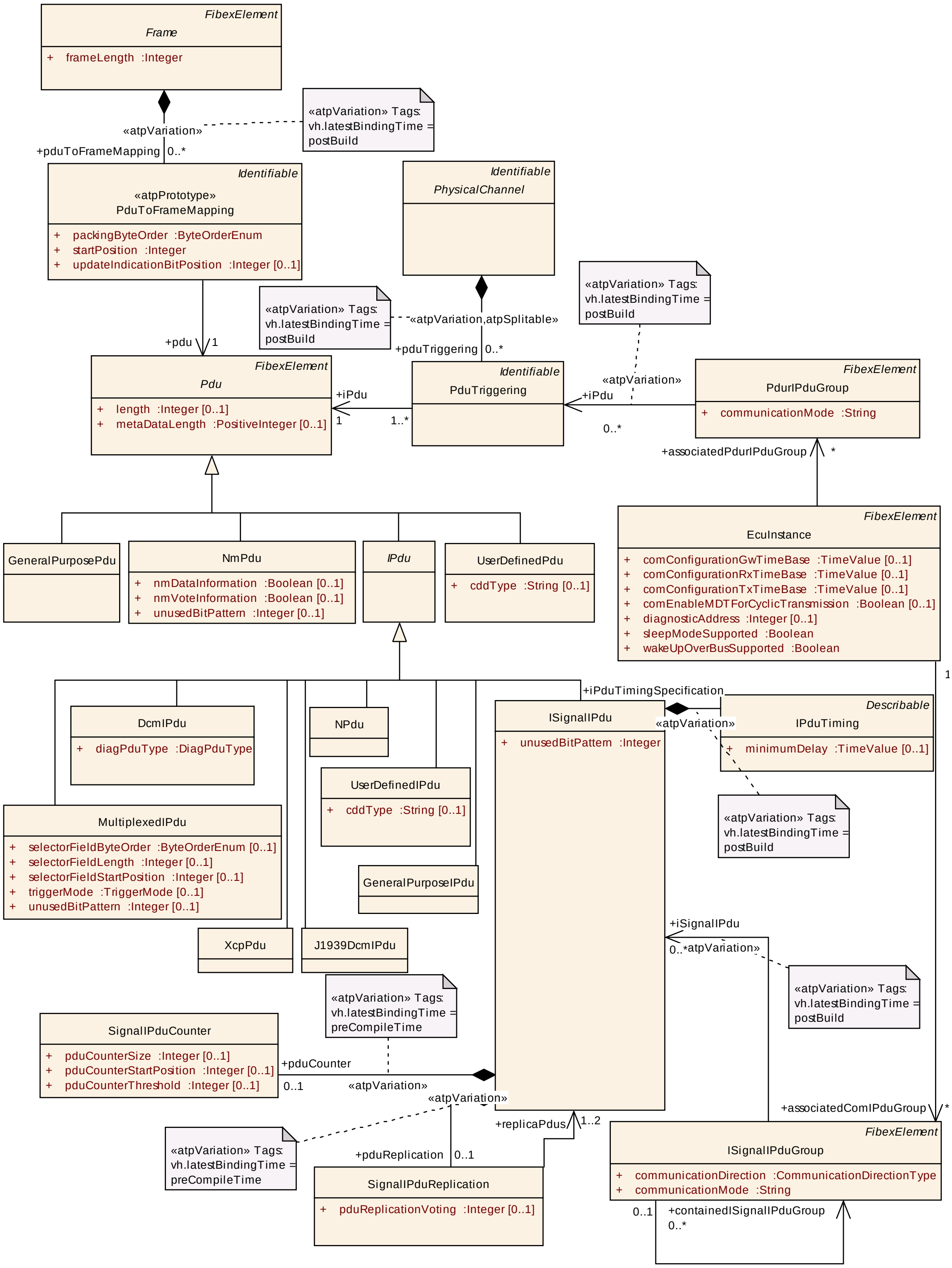

Figure 6.7: Pdus and the mapping into Frames (FibexCore: PDUOverview)

The PduToFrameMapping element describes the mapping of Pdus to Frames and defines the position of a Pdu within a Frame. By using different PduToFrameMappings it is possible to use the same Pdu in different Frames.

A timing description IPduTiming can be aggregated directly by the ISignalIPdu. This timing description can be used for the Configuration of COM Transmission Modes. The PduTriggering describes on which channel the Pdu is transmitted. Timing requirements may be specified with the Timing Extension model. More details are described in chapter 1.7.3. Such Pdu timing requirements needs to be fulfilled by the timing specification on the Frame.

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,Multilanguage Referrable,PackageableElement,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
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.
metaDataLength	PositiveInteger	0..1	attr	Number of additional bytes of MetaData in the PDU data field. The MetaData contains auxiliary information for the PDU, e.g. the CAN ID.

Table 6.18: Pdu

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,Multilanguage Referrable,PackageableElement,Pdu,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 6.19: IPdu

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,Multilanguage Referrable,PackageableElement,Pdu,Referrable			
Attribute	Datatype	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
iSignalToPduMapping	ISignalToIPduMapping	*	aggr	Definition of SignalToIPduMappings included in the SignalIPdu. 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 6.20: ISignalIPdu

Class	SignalIPduCounter			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	A PduCounter is included in a predefined set of Pdus and used to ensure that a sequence of Pdus is maintained. The counter is incremented when a Pdu is transmitted. The receivers check if the received Pdu is the next one in sequence.			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
pduCounterSize	Integer	0..1	attr	Size of PduCounter expressed in bits. Range: 1..8
pduCounterStartPosition	Integer	0..1	attr	Position of PduCounter expressed in bits. Note that PduCounter is not allowed to cross a byte border.
pduCounterThreshold	Integer	0..1	attr	Threshold value of IPduCounter algorithm. See AUTOSAR COM Spec for more details.

Table 6.21: SignalIPduCounter

Class	SignalIPduReplication			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	PduReplication 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.			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
pduReplicationVoting	Integer	0..1	attr	Number of identical IPdus needed for successful voting (1-3).
replicaPdus	ISignalIPdu	1..2	ref	Reference to replica Pdus of this IPdu.

Table 6.22: SignalIPduReplication

Class	NmPdu			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Network Management Pdu **Tags:** atp.recommendedPackage=Pdus			
Base	ARObject,CollectableElement,FibexElement,Identifiable,MultilanguageReferrable,PackageableElement,Pdu,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
iSignalToIPduMapping	ISignalToIPduMapping	*	aggr	This optional aggregation is used to describe NmUserData 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.

Attribute	Datatype	Mul.	Kind	Note
nmDataInformation	Boolean	0..1	attr	Defines if the Pdu contains NM Data. If the NmPdu does not aggregate any ISignalToIPduMappings it still may contain UserData that is set via Nm_SetUserData(). If the ISignalToIPduMapping exists then the nmDataInformation attribute shall be ignored.
nmVoteInformation	Boolean	0..1	attr	Defines if the Pdu contains NM Vote information.
unusedBitPattern	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 nmDataInformation attribute is set to true.

Table 6.23: NmPdu

[constr_3056] pduLength of the NmPdu ⌈ The pduLength of the NmPdu shall be restricted to 0..8. ⌋

Please note that in AUTOSAR only FrNm is able to send out NmPdus with and without voting information:

[constr_3073] nmVoteInformation only valid for FrNm ⌈ The nmVoteInformation attribute is only valid for FrNm. ⌋

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,PackageableElement,Pdu,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
–	–	–	–	–

Table 6.24: NPdu

Class	**XcpPdu**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	This element is deprecated and will be removed in future. The GeneralPurposeIPdu with the category "Xcp" shall be used instead. Old description: AUTOSAR XCP Pdu. **Tags:** atp.Status=obsolete; atp.StatusComment=The GeneralPurposeIPdu with the category "Xcp" shall be used instead.; atp.StatusRevisionBegin=4.1.2; atp.recommendedPackage=Pdus			
Base	ARObject,CollectableElement,FibexElement,IPdu,Identifiable,Multilanguage Referrable,PackageableElement,Pdu,Referrable			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
–	–	–	–	–

Table 6.25: XcpPdu

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,Multilanguage Referrable,PackageableElement,Pdu,Referrable			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
diagPduType	DiagPduType	1	attr	Attribute is used to distinguish a request from a response.

Table 6.26: DcmIPdu

Enumeration	**DiagPduType**
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::Core Communication
Note	Used to distinguish a diagnostic request from a response.
Literal	*Description*
diagRequest	Diagnostic Request
diagResponse	Diagnostic Response

Table 6.27: DiagPduType

Class	J1939DcmIPdu			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	This element is deprecated and will be removed in future. The GeneralPurposeIPdu with the category "J1939Dcm" shall be used instead. Old description: Represents the IPdus handled by J1939Dcm. **Tags:** atp.Status=obsolete; atp.StatusComment=The GeneralPurposeIPdu with the category "J1939Dcm" shall be used instead.; atp.StatusRevisionBegin=4.1.2; atp.recommendedPackage=Pdus			
Base	ARObject,CollectableElement,FibexElement,IPdu,Identifiable,Multilanguage Referrable,PackageableElement,Pdu,Referrable			
Attribute	Datatype	Mul.	Kind	Note
–	–	–	–	–

Table 6.28: J1939DcmIPdu

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,Multilanguage Referrable,PackageableElement,Pdu,Referrable			
Attribute	Datatype	Mul.	Kind	Note
–	–	–	–	–

Table 6.29: GeneralPurposePdu

[constr_3081] Value of category in GeneralPurposePdu ⌈ The attribute category of GeneralPurposePdu can have the following values:

- Sd (Service Discovery)

⌋

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,Multilanguage Referrable,PackageableElement,Pdu,Referrable			
Attribute	Datatype	Mul.	Kind	Note
–	–	–	–	–

Table 6.30: GeneralPurposeIPdu

[constr_3082] Value of category in GeneralPurposeIPdu ⌈ The attribute category of GeneralPurposeIPdu can have the following values:

- Xcp
- J1939Dcm

⌋

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,Multilanguage Referrable,PackageableElement,Pdu,Referrable			
Attribute	**Datatype**	**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 6.31: UserDefinedPdu

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,Multilanguage Referrable,PackageableElement,Pdu,Referrable			
Attribute	**Datatype**	**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 6.32: UserDefinedIPdu

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	**Datatype**	**Mul.**	**Kind**	**Note**

Attribute	Datatype	Mul.	Kind	Note
packingByteOrder	ByteOrderEnum	1	attr	This attribute defines the order of the bytes of the Pdu and the packing into the Frame. The byte ordering "Little Endian" (MostSignificantByteLast) and "Big Endian" (MostSignificantByteFirst) can be selected. A mix between Little Endian and Big Endian within a Frame is not allowed (all PduToFrameMappings within a Frame must have the same packingByteOrder).
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 packingByteOrder 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 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.
updateIndicationBitPosition	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 updateIndicationBitPosition is linked to the value of the attribute packingByteOrder because the method of finding the bit position is different for the values mostSignificantByteFirst and mostSignificantByteLast. This means that if the value of packingByteOrder is changed while the value of updateIndicationBitPosition remains unchanged the exact bit position of updateIndicationBitPosition within the enclosing Frame still undergoes a change.

Table 6.33: PduToFrameMapping

Class	**IPduTiming**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	AUTOSAR COM provides the possibility to define two different TRANSMISSION MODES for each IPdu. The Transmission Mode of an IPdu that is valid at a specific point in time is selected using the values of the signals that are mapped to this IPdu. For each IPdu a Transmission Mode Selector is defined. The Transmission Mode Selector is calculated by evaluating the conditions for a subset of signals (class TransmissionModeCondition in the System Template). The Transmission Mode Selector is defined to be true, if at least one Condition evaluates to true and is defined to be false, if all Conditions evaluate to false.			
Base	ARObject,Describable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
minimumDelay	TimeValue	0..1	attr	Minimum Delay in seconds between successive transmissions of this I-PDU, independent of the Transmission Mode.
transmissionModeDeclaration	TransmissionModeDeclaration	0..1	aggr	AUTOSAR COM allows configuring statically two different transmission modes for each I-PDU (True and False). The Transmission Mode Selector evaluates the conditions for a subset of signals and decides the transmission mode. It is possible to switch between the transmission modes during runtime.

Table 6.34: IPduTiming

Class	**PduTriggering**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	The PduTriggering describes on which channel the IPdu is transmitted. The Pdu routing by the PduR is only allowed for "IPdus" and not for NmPdus and XcpPdus. 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	***Datatype***	***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 "IPdus" and not for NmPdus and XcpPdus. Nevertheless is the reference to the Pdu element necessary since the PduTriggering element is also used to specify the sending and receiving connections to EcuPorts.

Attribute	Datatype	Mul.	Kind	Note
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.
iSignalTriggering	ISignalTriggering	*	ref	This reference provides the relationship to the ISignalTriggerings 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

Table 6.35: PduTriggering

AUTOSAR COM provides a mechanism of starting/stopping COM PDU groups (`ISignalIPduGroup`).

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. When an ISignalIPduGroup containing one or more other ISignalIPduGroups is started the containedISignalIPduGroups shall also be started. When an ISignalIPduGroup containing one or more other ISignalIPduGroups is stopped the contained ISignalIPduGroups shall also be stopped. Only a two level hierarchy of ISignalIPdu groups is allowed. An ISignalIPdu group that is part of an ISignalIPdu group must not contain ISignalIPduGroups. **Tags:** atp.recommendedPackage=ISignaliPduGroup			
Base	ARObject,CollectableElement,FibexElement,Identifiable,Multilanguage Referrable,PackageableElement,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
communicationDirection	CommunicationDirectionType	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).
communicationMode	String	1	attr	This attribute defines the use-case for this ISignalIPduGroup (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.
containedISignalIPduGroup	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.

Attribute	Datatype	Mul.	Kind	Note
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

Table 6.36: ISignalIPduGroup

Enumeration	CommunicationDirectionType
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication
Note	Describes the communication direction.
Literal	***Description***
in	Reception (Input)
out	Transmission (Output)

Table 6.37: CommunicationDirectionType

[constr_3020] communicationDirection of containedIPduGroups ⌈ The value of the attribute communicationDirection of containedIPduGroup must be identical to the value of the attribute communicationDirection of the enclosing ISignalIPduGroup. ⌋

The AUTOSAR Pdu Router provides a mechanism of enabling/disabling of routing path groups (PdurIPduGroup).

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,PackageableElement,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
communicationMode	String	1	attr	This attribute defines the use-case for this PduRIPduGroup. 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.

Attribute	Datatype	Mul.	Kind	Note
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 PduTriggering 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 6.38: PdurIPduGroup

6.3.1 EndToEndProtection for ISignalIPduGroups

[TPS_SYST_01070] E2E Protection of ISignalGroups ⌈ It is possible to protect the inter-ECU data exchange of safety-related ISignalGroups which are mapped into ISignalIPdus using protection mechanisms provided by E2E Library. ⌋*(RS_SYST_00028)*

[TPS_SYST_01071] E2E Protection of several ISignalGroups in one ISignalIPdu ⌈ It is possible to protect several ISignalGroups in one ISignalIPdu using several EndToEndProtectionISignalIPdu elements. ⌋*(RS_SYST_00028)*

The EndToEndProtectionISignalIPdu element refers to the ISignalGroup that is to be protected and to the ISignalIPdu that transmits the protected ISignalGroup. The dataOffset in the EndToEndProtectionISignalIPdu element defines the starting position of the Array representation of the ISignalGroup.

The information how the referenced ISignalGroup shall be protected (through which E2E Profile and with which E2E settings) is defined in the EndToEndDescription element.

[TPS_SYST_01072] Offset attributes of EndToEndDescription ⌈ All offset attributes of EndToEndDescription are relative to the dataOffset with respect to the ISignalIPdu (absolute position of the CRC = dataOffset + crcOffset). ⌋*(RS_SYST_00028)*

For more details, see End to End Library [18].

[TPS_SYST_01073] E2E Protection via COM Callouts ⌈ If the E2E Protection is done via COM Callouts then the EndToEndProtectionISignalIPdu shall be defined. ⌋*(RS_SYST_00028)*

[TPS_SYST_01074] E2E Protection in the E2E Wrapper ⌈ If the E2E Protection is done in the E2E Wrapper then both EndToEndProtectionISignalIPdu and EndToEndProtectionVariablePrototype shall be defined. ⌋*(RS_SYST_00028)*

For more details, see Software Component Template specification [5].

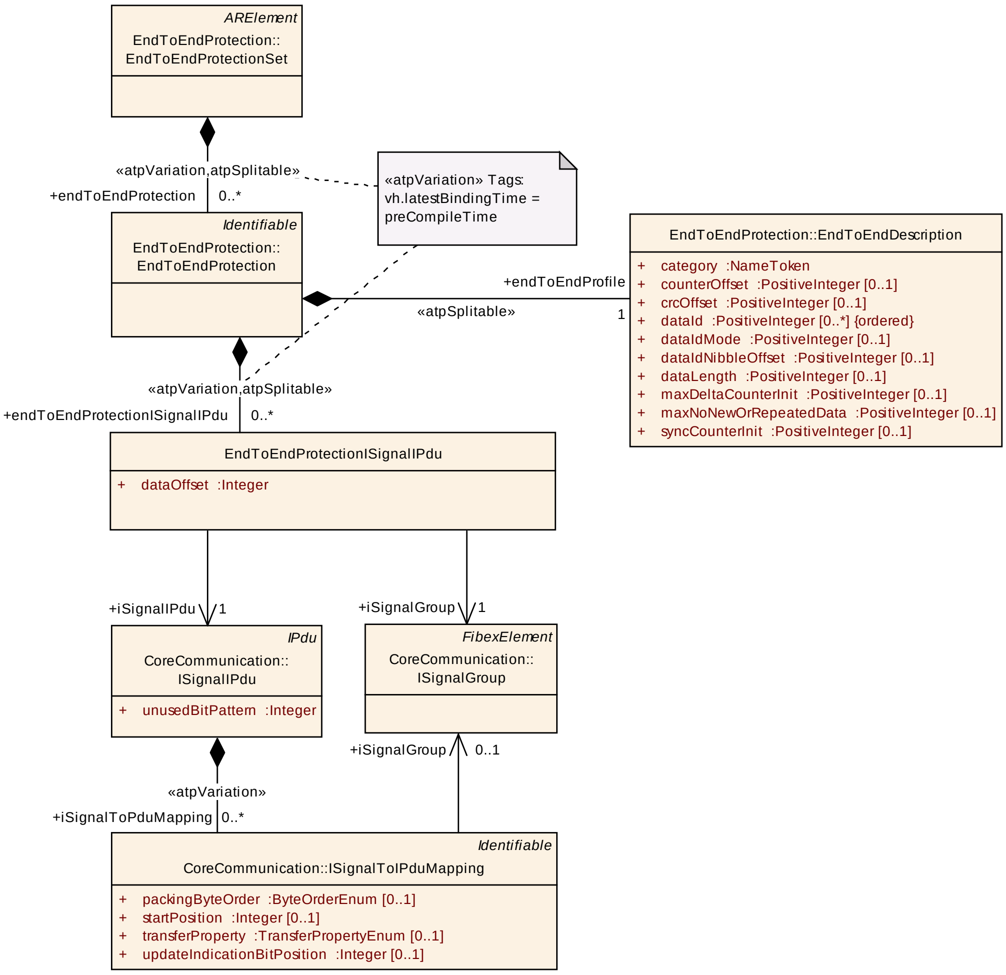

Figure 6.8: EndToEndProtection for COM IPdus

Class	**EndToEndProtectionSet**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::EndToEndProtection			
Note	This represents a container for collection EndToEndProtectionInformation. **Tags:** atp.recommendedPackage=EndToEndProtectionSets			
Base	ARElement,ARObject,CollectableElement,Identifiable,Multilanguage Referrable,PackageableElement,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
endToEnd Protection	EndToEndProte ction	*	aggr	This is one particular EndToEndProtection. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variation Point.shortLabel vh.latestBindingTime=preCompileTime

Table 6.39: EndToEndProtectionSet

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	**Datatype**	**Mul.**	**Kind**	**Note**
endToEndProfile	EndToEndDescription	1	aggr	This represents the particular EndToEndDescription. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=endToEndProfile
endToEndProtectionISignalIPdu	EndToEndProtectionISignalIPdu	*	aggr	Defines to which ISignalIPdu - ISignalGroup pair this EndToEndProtection shall apply. In case several ISignalGroups are used to transport the data (e.g. fan-out in the RTE) there may exist several EndToEndProtectionISignalIPdu definitions. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=variationPoint.shortLabel vh.latestBindingTime=preCompileTime
endToEndProtectionVariablePrototype	EndToEndProtectionVariablePrototype	*	aggr	Defines to which VariableDataPrototypes in the roles of one sender and one or more receivers this EndToEndprotection applies. It shall be possible to aggregate several EndToEndProtectionVariablePrototype 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 6.40: EndToEndProtection

Class	**EndToEndProtectionISignalIPdu**			
Package	M2::AUTOSARTemplates::SystemTemplate::EndToEndProtection			
Note	It is possible to protect the inter-ECU data exchange of safety-related ISignalGroups at the level of COM IPdus using protection mechanisms provided by E2E Library. For each ISignalGroup to be protected, a separate EndToEndProtectionISignalIPdu element shall be created within the EndToEndProtectionSet. The EndToEndProtectionISignalIPdu element refers to the ISignalGroup that is to be protected and to the ISignalIPdu that transmits the protected ISignalGroup. The information how the referenced ISignalGroup shall be protected (through which E2E Profile and with which E2E settings) is defined in the EndToEndDescription element.			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
dataOffset	Integer	1	attr	This attribute defines the beginning offset (in bits) of the Array representation of the Signal Group (including CRC, counter and application signal group) in the IPdu. This attribute is mandatory and the dataOffset shall always be defined.
iSignalGroup	ISignalGroup	1	ref	Reference to the ISignalGroup that is to be protected.
iSignalIPdu	ISignalIPdu	1	ref	Reference to the ISignalIPdu that transmits the protected ISignalGroup.

Table 6.41: EndToEndProtectionISignalIPdu

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	**Datatype**	**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

Attribute	Datatype	Mul.	Kind	Note
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 EndToEndProtection. **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
dataIdNibbleOffset	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

Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
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
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 MaxDeltaCounterInit is 1, then at the next reception the receiver can accept Counters with values 2 and 3, but not 4. Note that if the receiver does not receive new Data at a consecutive read, then the receiver increments the tolerance by 1. **Tags:** xml.sequenceOffset=-70
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. **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 6.42: EndToEndDescription

The maxDeltaCounterInit, maxNoNewOrRepeatedData and syncCounterInit values can also be specified in the ReceiverComSpec. This allows the definition of receiver specific values. Values for maxDeltaCounterInit, maxNoNewOrRepeatedData and syncCounterInit that are defined in the ReceiverComSpec override the possible values in the EndToEndDescription class. More details can be found in the Software Component Template specification [5].

The supported E2E profiles (possible values of category in EndToEndDescription) are described in the Software Component Template [5] and the End to End Library [18].

6.4 IPdu Timing

AUTOSAR COM allows configuring statically two different transmission modes for each IPdu (True and False). `TransmissionModeDeclaration` uses a transmission mode selector, calculated from a number of individual `TransmissionModeCondition`s or `ModeDrivenTransmissionModeCondition`s to decide which of the two modes is selected. It is possible to switch between the transmission modes during runtime.

Figure 6.9: IPdu Timing

[TPS_SYST_01075] Signal content evaluation via `TransmissionModeCondition` ⌈ The signal content can be evaluated as the transmission mode selector via the `TransmissionModeCondition`s. ⌋*(RS_SYST_00037)*

[TPS_SYST_01076] Mode evaluation via `modeDrivenTrueCondition` or `modeDrivenFalseCondition` ⌈ Mode conditions can be evaluated as the transmission mode selector via the modeDrivenTrueConditions or modeDrivenFalseConditions. ⌋*(RS_SYST_00037)*

[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. ⌋

To use the signal content evaluation a TransmissionModeCondition can be attached to each signal within an IPdu. Each TransmissionModeCondition contains a reference to a signal and to an assigned filter. The filter condition is used for the selection of the transmission mode. If at least one condition in the signal content evaluation is true, Transmission Mode "TRUE" shall be used for this IPdu. In all other cases, the Transmission Mode "FALSE" shall be used. More details can be found in the COM Specification [19].

[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. ⌋

In the mode driven evaluation ModeDeclarations are evaluated. The transmissionModeFalseTiming is activated if all defined modeDrivenFalseConditions evaluate to true and the transmissionModeTrueTiming is activated if all defined modeDrivenTrueConditions evaluate to true. Each condition that is defined by ModeDrivenTransmissionModeCondition evaluates to true if one of the referenced ModeDeclarations is active.

The TransmissionModeDeclaration element aggregates the TransmissionModeTiming in two different roles: transmissionModeTrueTiming and transmissionModeFalseTiming. The available COM Transmission Mode Timings can be described by the CyclicTiming and EventControlledTiming elements (see Table 6.43) that are aggregated by the TransmissionModeTiming class.

[TPS_SYST_01077] Mapping of Com Transmission Modes to System Template elements ⌈

COM Transmission Modes	*Description*	*realization in System Template*
Periodic	Transmissions occur indefinitely with a fixed period between them	CyclicTiming
Direct/n-times	Event driven transmission with n-1 repetitions	EventControlledTiming
Mixed	Periodic transmission with direct/n-times transmissions in between	EventControlledTiming and CyclicTiming
None	No transmission	no timing assigned

Table 6.43: COM Transmission Modes

⌋(*RS_SYST_00037*)

Class	**TransmissionModeDeclaration**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing			
Note	AUTOSAR COM provides the possibility to define two different TRANSMISSION MODES (True and False) for each I-PDU. As TransmissionMode selector the signal content can be evaluated via transmissionModeCondition (implemented directly in the COM module) or mode conditions can be defined with the modeDrivenTrueCondition or modeDrivenFalseCondition (evaluated by BswM and invoking Com_SwitchIpduTxMode COM API). If modeDrivenTrueCondition and modeDrivenFalseCondition are defined they shall never evaluate to true both at the same time. The mixing of Transmission Mode Switch via API and signal value is not allowed.			
Base	ARObject			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
modeDrivenFalseCondition	ModeDrivenTransmissionModeCondition	*	aggr	Defines the trigger for the Com_SwitchIpduTxMode Transmission Mode switch. Only if all defined modeDrivenFalseConditions evaluate to true (AND associated) the transmissionModeFalseTiming shall be activated. modeDrivenTrueCondition and modeDrivenFalseCondition shall never evaluate to true both at the same time.
modeDrivenTrueCondition	ModeDrivenTransmissionModeCondition	*	aggr	Defines the trigger for the Com_SwitchIpduTxMode Transmission Mode switch. Only if all defined modeDrivenTrueConditions evaluate to true (AND associated) the transmissionModeTrueTiming shall be activated. modeDrivenTrueCondition and modeDrivenFalseCondition shall never evaluate to true both at the same time.
transmissionModeCondition	TransmissionModeCondition	*	aggr	The Transmission Mode Selector evaluates the conditions for a subset of signals and decides which transmission mode should be used. In case only one transmission mode is used there is no need for the "TransmissionModeCondition" and its sub-structure. In case the transmission mode shall be switched using the COM-API "Com_SwitchIpduTxMode" there is no need for the´"TransmissionModeCondition" and its sub-structure.
transmissionModeFalseTiming	TransmissionModeTiming	0..1	aggr	Timing Specification if the COM Transmission Mode is false. The Transmission Mode Selector is defined to be false, if all Conditions evaluate to false.
transmissionModeTrueTiming	TransmissionModeTiming	0..1	aggr	Timing Specification if the COM Transmission Mode is true. The Transmission Mode Selector is defined to be true, if at least one Condition evaluates to true.

Attribute	Datatype	Mul.	Kind	Note

Table 6.44: TransmissionModeDeclaration

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	*Datatype*	*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 6.45: TransmissionModeCondition

Class	**ModeDrivenTransmissionModeCondition**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing			
Note	The condition defined by this class evaluates to true if one of the referenced modeDeclarations (OR associated) is active. All referenced modeDeclarations shall be from the same ModeDeclarationGroup. The condition is used to define which TransmissionMode shall be activated using Com_SwitchIpduTxMode.			
Base	ARObject			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
modeDeclaration	ModeDeclaration	1..*	ref	Reference to one modeDeclaration which is OR associated in the context of the ModeDrivenTransmissionModeCondition.

Table 6.46: ModeDrivenTransmissionModeCondition

The `ModeDeclaration` and the `ModeDeclarationGroup` is described in more detail in the Software Component Template Specification [5].

Class	**TransmissionModeTiming**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing			
Note	If the COM Transmission Mode is false the timing is aggregated by the TransmissionModeTiming element in the role of transmissionModeFalseTiming. If the COM Transmission Mode is true the timing is aggregated by the TransmissionModeTiming element in the role of transmissionModeTrueTiming. COM supports the following Transmission Modes: Periodic (Cyclic Timing) Direct /n-times (EventControlledTiming) Mixed (Cyclic and EventControlledTiming are assigned) None (no timing is assigned)			
Base	ARObject			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
cyclicTiming	CyclicTiming	0..1	aggr	Periodic Transmission Mode.
eventControlledTiming	EventControlledTiming	0..1	aggr	Direct Transmission Mode.

Table 6.47: TransmissionModeTiming

6.4.1 Data Filter configuration

Data Filters are used on sender side to configure Transmission Mode Conditions (TMC). On receiver side Data Filters can be used as filtering mechanisms for signals (see `ISignalPort` element). More details about the usage of `DataFilter`s can be found in the Software Component Template Specification [5].

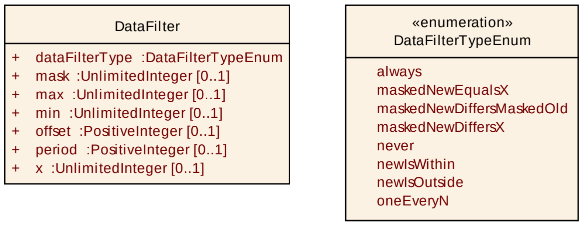

Figure 6.10: Data Filter

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	***Datatype***	***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
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 6.48: DataFilter

Enumeration	**DataFilterTypeEnum**
Package	M2::AUTOSARTemplates::CommonStructure::Filter
Note	This enum specifies the supported DataFilterTypes.
Literal	***Description***
always	No filtering is performed so that the message always passes.
maskedNewDiffersMaskedOld	Pass messages where the masked value has changed. (new_value&mask) !=(old_value&mask) new_value: current value of the message old_value: last value of the message (initialized with the initial value of the message, updated with new_value if the new message value is not filtered out)
maskedNewDiffersX	Pass messages whose masked value is not equal to a specific value x (new_value&mask) != x new_value: current value of the message

maskedNewEqualsX	Pass messages whose masked value is equal to a specific value x (new_value&mask) == x new_value: current value of the message
never	The filter removes all messages.
newIsOutside	Pass a message if its value is outside a predefined boundary. (min > new_value) OR (new_value > max)
newIsWithin	Pass a message if its value is within a predefined boundary. min <= new_value <= max
oneEveryN	Pass a message once every N message occurrences. Algorithm: occurrence % period == offset Start: occurrence = 0. Each time the message is received or transmitted, occurrence is incremented by 1 after filtering. Length of occurrence is 8 bit (minimum).

Table 6.49: DataFilterTypeEnum

6.4.2 Cyclic Timing

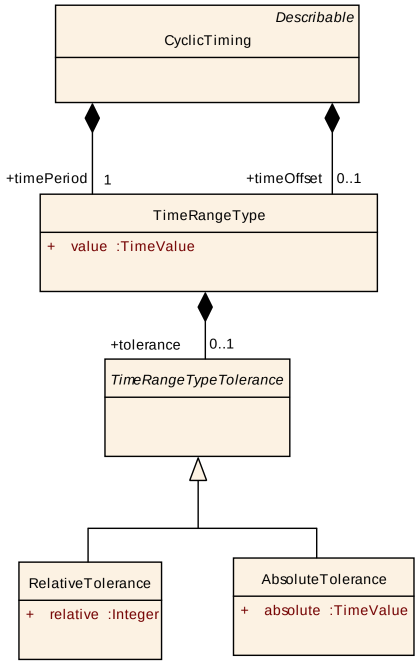

Figure 6.11: Cyclic Timing

Class	**CyclicTiming**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing			
Note	Specification of a cyclic sending behavior.			
Base	ARObject,Describable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
timeOffset	TimeRangeType	0..1	aggr	This attribute specifies the time until first transmission of this I-PDU. This attribute defines the time between Com_IpduGroupStart and the first transmission of the cyclic part of this transmission request for this I-PDU.
timePeriod	TimeRangeType	1	aggr	Period of the repetition of cyclic transmissions.

Table 6.50: CyclicTiming

6.4.3 EventControlled Timing

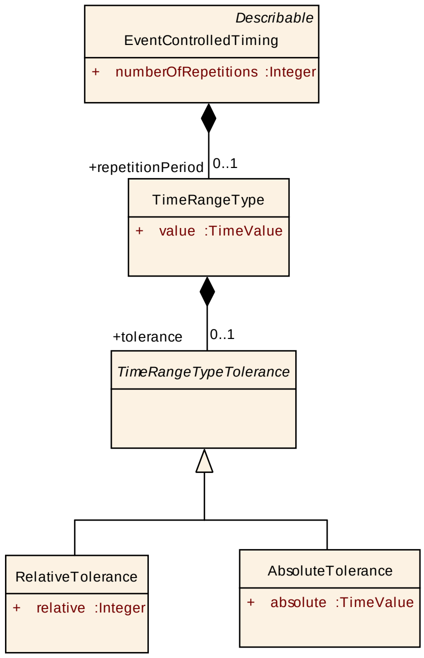

Figure 6.12: EventControlled Timing

Class	**EventControlledTiming**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing			
Note	Specification of a event driven sending behavior. The PDU is sent n (numberOfRepeat + 1) times separated by the repetitionPeriod. If numberOfRepeats = 0, then the Pdu is sent just once.			
Base	ARObject,Describable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**

Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
numberOf Repetitions	Integer	1	attr	Defines the number of repetitions for the Direct/N-Times transmission mode and the event driven part of Mixed transmission mode.
repetitionP eriod	TimeRangeTyp e	0..1	aggr	The repetitionPeriod specifies the time in seconds that elapses before the pdu can be sent the next time (Minimum repeat gap between two pdus). The repetitionPeriod is optional in case that no repetitions are configured.

Table 6.51: EventControlledTiming

Class	**TimeRangeType**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing			
Note	The timeRange can be specified with the value attribute. Optionally a tolerance can be defined.			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
tolerance	TimeRangeTyp eTolerance	0..1	aggr	Optional specification of a tolerance.
value	TimeValue	1	attr	Average value of a date (in seconds)

Table 6.52: TimeRangeType

Class	**RelativeTolerance**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing			
Note	Maximum allowable deviation			
Base	ARObject,TimeRangeTypeTolerance			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
relative	Integer	1	attr	Maximum allowable deviation in percent

Table 6.53: RelativeTolerance

Class	**AbsoluteTolerance**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing			
Note	Maximum allowable deviation			
Base	ARObject,TimeRangeTypeTolerance			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
absolute	TimeValue	1	attr	Maximum allowable deviation in duration (in seconds)

Table 6.54: AbsoluteTolerance

6.5 I-Pdu Multiplexer

Multiplexing is used to transport varying Com IPdus at the same position in a single multiplexed IPdu. A multiplexed IPdu consists of a dynamic part, a selector field and an optional static part. According to the value of the selector field the dynamic part can have a different layout.

[TPS_SYST_01078] Dynamic Part of a MultiplexedIPdu ⌈ For each alternative of a MultiplexedIPdu there is exactly one Com IPdu that is transmitted in the dynamic part. ⌋

[TPS_SYST_01079] Static Part of a MultiplexedIPdu ⌈ The static part of a MultiplexedIPdu is the same regardless of the selector field and consists of exactly one Com IPdu. ⌋

The MultiplexedIPdu element contains attributes that describe the position and the length of a selector within an IPdu. A selector is a bitfield of certain length, by the value of which the corresponding data region of the dynamic part must be interpreted dynamically, i.e. at run-time.

[constr_3007] selectorFieldCodes for dynamic part alternatives ⌈ The selectorFieldCodes for the dynamic part alternatives within one MultiplexedIPdu shall differ from each other. ⌋

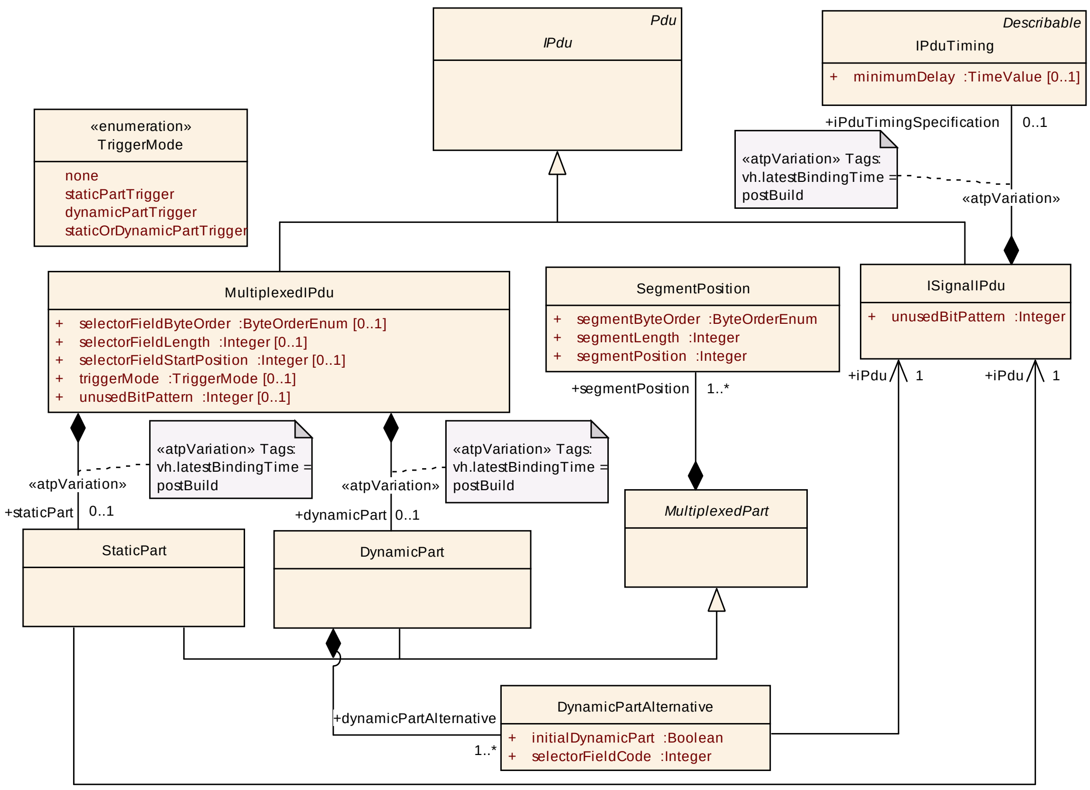

Figure 6.13: I-Pdu Multiplexer (FibexCore: IPDUMultiplexerOverview)

Enumeration	**TriggerMode**
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::Core Communication
Note	IPduM can be configured to send a transmission request for the new multiplexed I-PDU to the PDU-Router because of conditions/ modes.
Literal	***Description***
dynamicPart Trigger	IPduM sends a transmission request to the PduR if a dynamic part is received.
none	IPduM does not trigger transmission because of receiving anything of this IPdu in case of TriggerTransmit.
staticOrDy-namicPart Trigger	IPduM sends a transmission request to the PduR if a static or dynamic part is received.
staticPart Trigger	IPduM sends a transmission request to the PduR if a static part is received.

Table 6.55: TriggerMode

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 selectorField. 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,Multilanguage Referrable,PackageableElement,Pdu,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***

Attribute	Datatype	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
selectorFieldByteOrder	ByteOrderEnum	0..1	attr	This attribute defines the order of the bytes of the selectorField and the packing into the MultiplexedIPdu. The byte ordering "Little Endian" (MostSignificantByteLast) and "Big Endian" (MostSignificantByteFirst) can be selected. A mix between Little Endian and Big Endian within a MultiplexedIPdu (staticPart, dynamicPart, selectorField) is not allowed. 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.
selectorFieldLength	Integer	0..1	attr	The size in bits of the selector field shall be configurable in a range of 0-2031 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.

Attribute	Datatype	Mul.	Kind	Note
selectorFieldStartPosition	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". 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.

Attribute	Datatype	Mul.	Kind	Note
unusedBitPattern	Integer	0..1	attr	AUTOSAR COM and AUTOSAR IPDUM are filling not used areas of an IPdu with this bit-pattern. This attribute is mandatory to avoid undefined behavior. This byte-pattern will be repeated throughout the IPdu. In a complete System Description this attribute is mandatory. If a MultiplexedPdu is received by a Pdu Gateway and is not delivered to the IPduM but routed directly to a bus interface then the content of the MulitplexedPdu doesn't need to be described in the System Extract/Ecu Extract. To support this use case the multiplicity is set to 0..1.

Table 6.56: MultiplexedIPdu

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	**Datatype**	**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 6.57: StaticPart

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	**Datatype**	**Mul.**	**Kind**	**Note**
dynamicPartAlternative	DynamicPartAlternative	1..*	aggr	Com IPdu alternatives that are transmitted in the Dynamic Part of the MultiplexedIPdu.

Table 6.58: 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	**Datatype**	**Mul.**	**Kind**	**Note**

Attribute	Datatype	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.
initialDynamicPart	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.
selectorFieldCode	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 6.59: DynamicPartAlternative

Class	**MultiplexedPart (abstract)**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	The StaticPart and the DynamicPart have common properties. Both can be separated in multiple segments within the multiplexed PDU.			
Base	ARObject			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
segmentPosition	SegmentPosition	1..*	aggr	The StaticPart and the DynamicPart can be separated in multiple segments within the multiplexed PDU. Therefore the StaticPart and the DynamicPart can contain multiple SegmentPositions.

Table 6.60: MultiplexedPart

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	**Datatype**	**Mul.**	**Kind**	**Note**
segmentByteOrder	ByteOrderEnum	1	attr	This attribute defines the order of the bytes of the segment and the packing into the MultiplexedIPdu. The byte ordering "Little Endian" (MostSignificantByteLast) and "Big Endian" (MostSignificantByteFirst) can be selected. A mix between Little Endian and Big Endian within a MultiplexedIPdu (staticPart, dynamicPart, selectorField) is not allowed.
segmentLength	Integer	1	attr	Data Length of the segment in bits.
segmentPosition	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".

Table 6.61: SegmentPosition

Figure 6.14 shows an example of an IPdu Multiplexer. The static part of the multiplexed IPdu contains ComIPduA. The value of the selector field in the dynamic part decides which content is transmitted. ComIPduB is transmitted if the selector field value is "0". ComIPduC is transmitted if the selector field value is "1".

The static and the dynamic part can consist of more than one element. These sub parts of the static or dynamic parts are called segments. In Figure 6.14 the dynamic Part is segmented into two parts. More details can be found in [20].

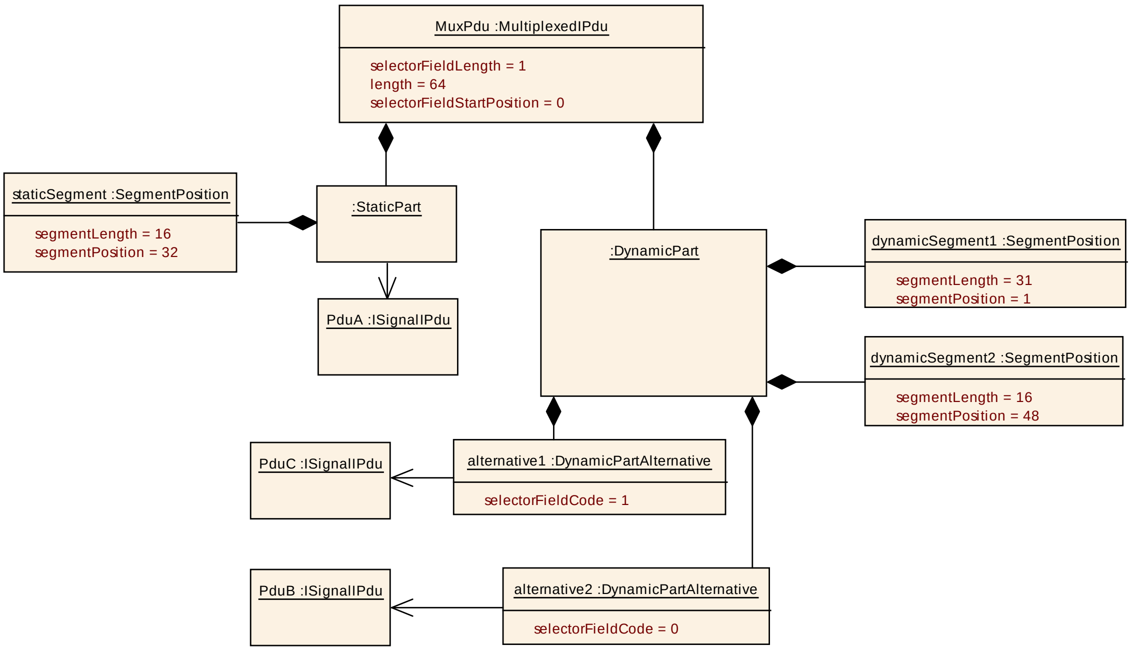

Figure 6.14: I-Pdu Multiplexer Example

[constr_3017] Length of multiplexed Pdu shall not be exceeded. ⌈The sum of included IPdus (static Part and dynamic Part) plus the length of the switch shall be smaller or equal than the length of the containing multiplexer Pdu. ⌋

6.5.1 I-Pdu Multiplexer in System Extract/ECU Extract

The processing in the ECU determines the description of MultiplexedIPdus in the System Extract/Ecu Extract. In case that a Gateway ECU only routes a MultiplexedIPdu without being interested in the content leads to a reduced description in the System Extract/ECU Extract. The following items describe the different scenarios and the consequences for the System Extract/ECU Extract description. A complete System Description contains all information.

[TPS_SYST_01080] Sending or receiving of a MultiplexedIPdu in System Extract/ECU Extract ⌈

- all attributes of the MultiplexedIPdu are mandatory
- aggregated DynamicPart with associated ISignalIPdus is mandatory in case
  - of sending
  - of receiving if at least one DynamicPartAlternative is received by one Ecu of the Extract.
- a PduTriggering shall be defined for the MultiplexedIPdu
- a PduTriggering shall be defined for all included ISignalIPdus in the DynamicPart and StaticPart

⌋

The initial ECU Configuration Generator configures COM, PduR, IpduM and lower layers with the information from the System Extract/ECU Extract.

[TPS_SYST_01081] Gatewaying of a MultiplexedIPdu in System Extract/ECU Extract ⌈

- StaticPart and DynamicPart definitions shall be omitted, thus no ISignalIPdu description shall be included
- all attributes of the MultiplexedIPdu shall be omitted.
- a PduTriggering shall be defined only for the gatewayed MultiplexedIPdu
- an IPduMapping between the source and the target PduTriggerings shall be defined

⌋

The initial ECU Configuration Generator configures PduR and lower layers with the information from the System Extract/ECU Extract.

[TPS_SYST_01082] Receiving and gatewaying of a MultiplexedIPdu in System Extract/ECU Extract ⌈

- all attributes of the MultiplexedIPdu are mandatory
- aggregated DynamicPart with associated ISignalIPdus is mandatory in case at least one DynamicPartAlternative is received by one Ecu of the Extract.
- a PduTriggering shall be defined for the MultiplexedIPdu
- an IPduMapping between the source and the target PduTriggerings shall be defined
- a PduTriggering shall be defined for all included ISignalIPdus in the DynamicPart and StaticPart

⌋

The initial ECU Configuration Generator configures Com, PduR, IpduM and lower layers with the information from the System Extract/ECU Extract.

6.6 Frames

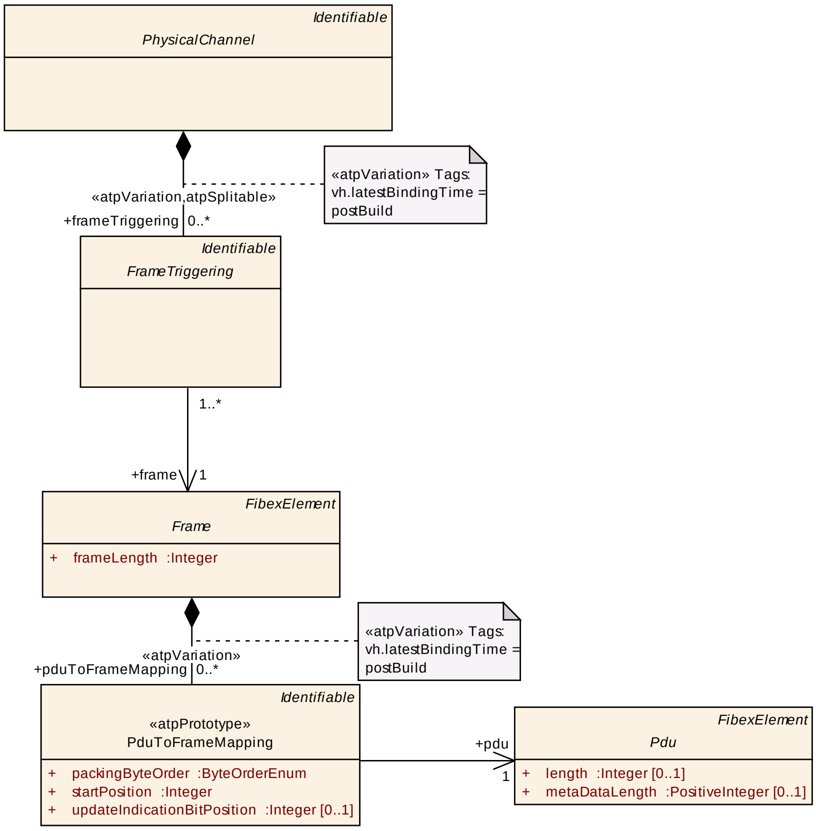

Figure 6.15: Frame Overview (FibexCore: FrameOverview)

[TPS_SYST_01083] Frame ⌈ A Frame represents a general design object that is used to describe the layout of the included Pdus as a reusable asset. ⌋

[TPS_SYST_01084] FrameTriggering ⌈ The FrameTriggering implements the reusable definition of a Frame within a concrete context and thus defines a Frame's send behavior and identification on a certain PhysicalChannel. ⌋

[constr_3012] Overlapping of Pdus is prohibited ⌈ Pdus mapped to a Frame shall NOT overlap. ⌋

[constr_3013] Frame length shall not be exceeded ⌈ The combined length of all Pdus that are mapped into a Frame shall not exceed the defined Frame length. ⌋

[constr_3014] Overlapping of updateIndicationBits for Pdus is prohibited ⌈ The updateIndicationBitPosition for a Pdu in a Frame shall NOT overlap with other updateIndicationBitPositions and Pdu locations. ⌋

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,PackageableElement,Referrable			
Attribute	Datatype	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.
pduToFrameMapping	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 6.62: Frame

Class	FrameTriggering (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	The FrameTriggering describes the instance of a frame sent on a channel and defines the manner of triggering (timing information) and identification of a frame on the channel, on which it is sent. For the same frame, if FrameTriggerings exist on more than one channel of the same cluster the fan-out/in is handled by the Bus interface.			
Base	ARObject,Identifiable,MultilanguageReferrable,Referrable			
Attribute	Datatype	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 PduTriggerings 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 6.63: FrameTriggering

6.7 Specialized Attributes of the Communication Entities

In the Basic Software the timing of bus frames can be controlled by send requests of the RTE in combination with the Transmission Mode and Transfer Property parameters in COM. On the other hand the timing can be controlled by the FlexRay Interface and LIN Interface.

This chapter describes the protocol specific extensions to the communication elements.

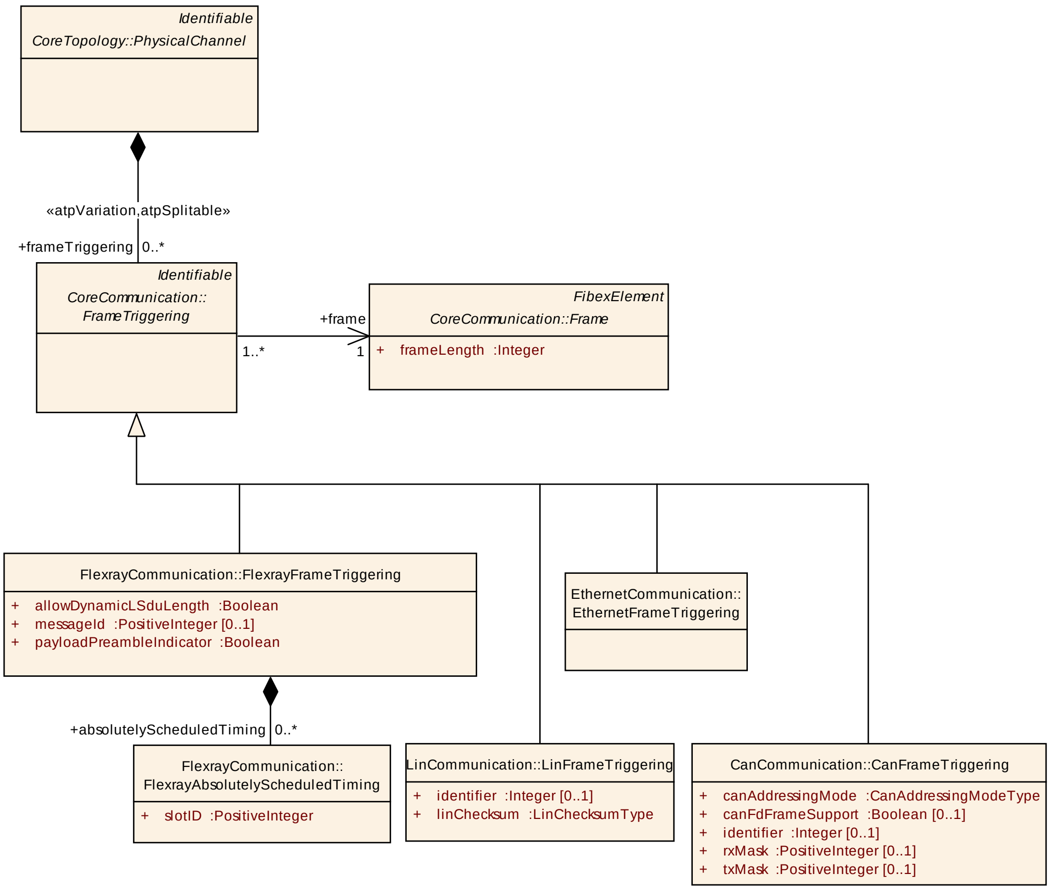

Figure 6.16: Frame Triggering

6.7.1 FlexRay specific description

[TPS_SYST_01128] Communication over FlexRay ⌈ The System Template supports the description of communication over FlexRay. ⌋*(RS_SYST_00024)*

In the following, the elements necessary to describe the FlexRay communication are specified.

FlexRay static segment parameters: Each `FlexrayFrameTriggering` is identified by its `slotId` and `communicationCycle`. In the static segment all communication slots are of identical, statically configured duration and all `FrameTriggering`s are of identical, statically configured length.

The sending behavior where the exact time for the FlexrayFrameTriggerings transmission is guaranteed is provided in the System Template by the usage of FlexrayAbsolutelyScheduledTiming.

In the cycle counter field of every frame, the current value of the cycle counter is transmitted (see FlexRay frame format). This value is incremented at the beginning of each new cycle, ranging from 0 to 63, and is reset to 0 after a sequence of 64 cycles.

[TPS_SYST_01085] Transmission of a FrameTriggering multiple times within one communication cycle ⌈ In the static segment FlexrayFrameTriggerings can be sent multiple times within one communication cycle. For describing this case multiple FlexrayAbsolutelyScheduledTimings shall be used. ⌋*(RS_SYST_00024)*

FlexRay dynamic segment parameters: In the dynamic segment the duration of communication slots may vary in order to accommodate frames of varying length. Furthermore, in the dynamic part, the slotId is equivalent to a priority. The higher the number the lower is the priority.

The frames in the static and in the dynamic segment are described in the same way. Each FlexrayFrameTriggering is identified by its slotId and communicationCycle. A description is provided by the usage of FlexrayAbsolutelyScheduledTiming.

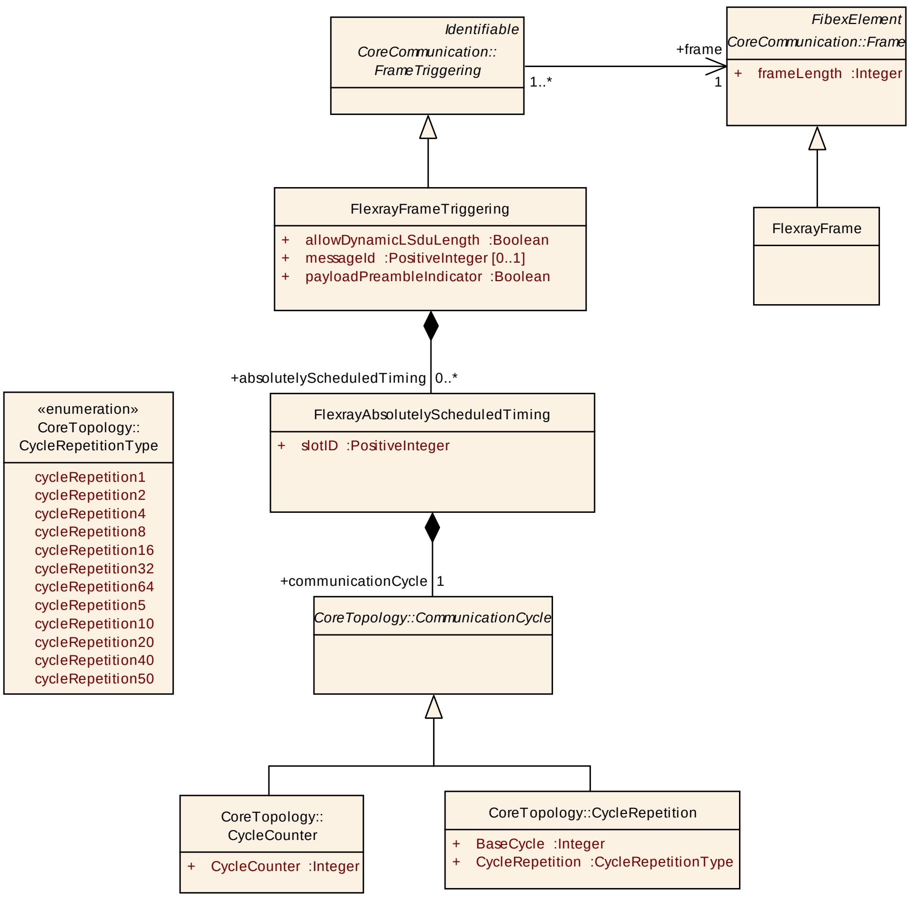

Figure 6.17: FlexRay Absolutely Scheduled Timing (Fibex4FlexRay:FlexrayAbsolutelyScheduledTiming)

Class	FlexrayFrame			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Flexray::Flexray Communication			
Note	FlexRay specific Frame element. **Tags:** atp.recommendedPackage=Frames			
Base	ARObject,CollectableElement,FibexElement,Frame,Identifiable,Multilanguage Referrable,PackageableElement,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 6.64: FlexrayFrame

Class	FlexrayFrameTriggering			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Flexray::Flexray Communication			
Note	FlexRay specific attributes to the FrameTriggering			
Base	ARObject,FrameTriggering,Identifiable,MultilanguageReferrable,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
absolutely Scheduled Timing	FlexrayAbsolute lyScheduledTim ing	*	aggr	Specification of a sending behaviour where the exact time for the frames transmission is guaranteed.
allowDyna micLSduLe ngth	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.
payloadPr eambleIndi cator	Boolean	1	attr	Switching the Payload Preamble bit.

Table 6.65: FlexrayFrameTriggering

Class	**FlexrayAbsolutelyScheduledTiming**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Flexray::Flexray Communication			
Note	Each frame in FlexRay is identified by its slot id and communication cycle. A description is provided by the usage of AbsolutelyScheduledTiming. In the static segment a frame can be sent multiple times within one communication cycle. For describing this case multiple AbsolutelyScheduledTimings have to be used. The main use case would be that a frame is sent twice within one communication cycle.			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
communicationCycle	CommunicationCycle	1	aggr	The communication cycle where the frame is sent.
slotID	PositiveInteger	1	attr	In the static part the SlotID defines the slot in which the frame is transmitted. The SlotID also determines, in combination with FlexrayCluster::numberOfStaticSlots, whether the frame is sent in static or dynamic segment. In the the dynamic part, the slot id is equivalent to a priority. Lower dynamic slot ids are all sent until the end of the dynamic segment. Higher numbers, which were ignored that time, have to wait one cycle and then must try again. minValue: 1 maxValue: 2047

Table 6.66: FlexrayAbsolutelyScheduledTiming

Class	**CommunicationCycle (abstract)**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology			
Note	The communication cycle where the frame is sent.			
Base	ARObject			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
–	–	–	–	–

Table 6.67: CommunicationCycle

The communication cycle can be described by the `CycleCounter` or by the `CycleRepetition`:

Class	**CycleCounter**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology			
Note	The communication cycle where the frame is send is described by the attribute "cycleCounter".			
Base	ARObject,CommunicationCycle			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
CycleCounter	Integer	1	attr	The communication cycle where the frame described by this timing is sent. If a timing is given in this way the referencing cluster must specify the NUMBER-OF-CYCLES as upper bound and point of total repetition. This value is incremented at the beginning of each new cycle, ranging from 0 to 63, and is reset to 0 after a sequence of 64 cycles.

Table 6.68: CycleCounter

Class	**CycleRepetition**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology			
Note	The communication cycle where the frame is send is described by the attributes baseCycle and cycleRepetition.			
Base	ARObject,CommunicationCycle			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
BaseCycle	Integer	1	attr	The first communication cycle where the frame is sent. This value is incremented at the beginning of each new cycle, ranging from 0 to 63, and is reset to 0 after a sequence of 64 cycles.
CycleRepetition	CycleRepetitionType	1	attr	The number of communication cycles (after the first cycle) whenever the frame described by this timing is sent again.

Table 6.69: CycleRepetition

Enumeration	**CycleRepetitionType**
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology
Note	The number of communication cycles (after the first cycle) whenever the frame is sent again. The FlexRay communication controller allows only determined values.
Literal	*Description*
cycleRepetition1	Attribute cycleRepetition value="1" valid only for FlexRay Protocol 2.1 Rev A
cycleRepetition10	Attribute cycleRepetition value="10" to support FlexRay 3.0
cycleRepetition16	Attribute cycleRepetition value="16" valid only for FlexRay Protocol 2.1 Rev A
cycleRepetition2	Attribute cycleRepetition value="2" valid only for FlexRay Protocol 2.1 Rev A
cycleRepetition20	Attribute cycleRepetition value="20" to support FlexRay 3.0
cycleRepetition32	Attribute cycleRepetition value="32" valid only for FlexRay Protocol 2.1 Rev A
cycleRepetition4	Attribute cycleRepetition value="4" valid only for FlexRay Protocol 2.1 Rev A
cycleRepetition40	Attribute cycleRepetition value="40" to support FlexRay 3.0
cycleRepetition5	Attribute cycleRepetition value="5" to support FlexRay 3.0
cycleRepetition50	Attribute cycleRepetition value="50" to support FlexRay 3.0
cycleRepetition64	Attribute cycleRepetition value="64" valid only for FlexRay Protocol 2.1 Rev A
cycleRepetition8	Attribute cycleRepetition value="8" valid only for FlexRay Protocol 2.1 Rev A

Table 6.70: CycleRepetitionType

6.7.2 LIN specific description

LIN is a protocol that is based on a single master - multiple slave principle. In the following, the parameters will be specified, which are necessary to describe the LIN Schedule Tables and the LIN Frames.

[TPS_SYST_01129] Communication over LIN ⌈ The System Template supports the description of communication over LIN. ⌋ *(RS_SYST_00022)*

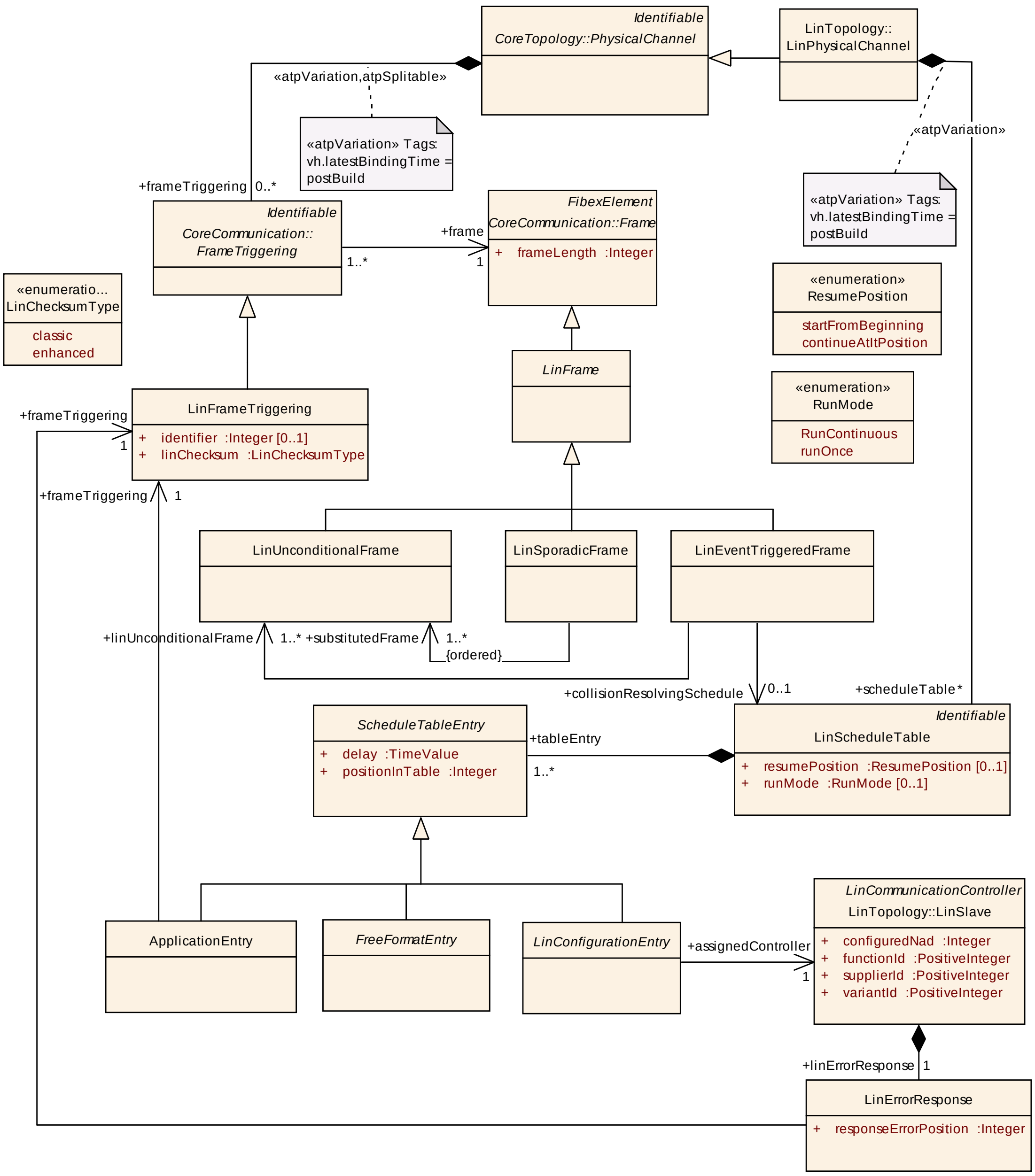

Figure 6.18: LIN Schedule Table (Fibex4Lin:LinScheduleTable)

6.7.2.1 LIN Frames

One LIN Frame consists of two parts: header and response. The header is always sent by a `LinMaster`, while the response is sent by only one dedicated `LinSlave`. There are three different ways of transmitting frames on the bus: unconditional, event triggered, and sporadic frames.

Class	**LinFrame (abstract)**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	Lin specific Frame element.			
Base	ARObject,CollectableElement,FibexElement,Frame,Identifiable,Multilanguage Referrable,PackageableElement,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 6.71: LinFrame

Class	**LinFrameTriggering**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	LIN specific attributes to the FrameTriggering			
Base	ARObject,FrameTriggering,Identifiable,MultilanguageReferrable,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
identifier	Integer	0..1	attr	To describe a frames identifier on the communication system, usually with a fixed identifierValue. For LinSporadicFrames the attribute shall be ignored.
linChecksum	LinChecksumType	1	attr	Type of checksum that the frame is using.

Table 6.72: LinFrameTriggering

Enumeration	**LinChecksumType**
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication
Note	Use of classic or enhanced checksum is managed by the master node and it is determined per frame identifier;
Literal	***Description***
classic	Classic in communication with LIN 1.3 slave nodes
enhanced	Enhanced in communication with LIN 2.0 slave nodes.

Table 6.73: LinChecksumType

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,Lin Frame,MultilanguageReferrable,PackageableElement,Referrable			
Attribute	Datatype	Mul.	Kind	Note
–	–	–	–	–

Table 6.74: LinUnconditionalFrame

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,Lin Frame,MultilanguageReferrable,PackageableElement,Referrable			
Attribute	Datatype	Mul.	Kind	Note
substitutedFrame (ordered)	LinUnconditionalFrame	1..*	ref	Reference to a group of unconditional frames that share the same frame slot. In case that more than one of the declared frames needs to be transferred, the one first listed shall be chosen. Within a channel a LIN Frame shall be referenced by only one FrameTriggering. This allows a derivation of the identifier of a substituted Frame. The identifier is specified in FrameTriggering element. A LinUnconditionalFrame associated with a LinSporadicFrame may not be allocated in the same LinScheduleTable as the sporadic frame.

Table 6.75: LinSporadicFrame

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	**Datatype**	**Mul.**	**Kind**	**Note**
collisionResolvingSchedule	LinScheduleTable	0..1	ref	Reference to the schedule table, which resolves a collision.
linUnconditionalFrame	LinUnconditionalFrame	1..*	ref	A list of slaves can respond to the master request if at least one of the signals carried in its unconditional frame is updated. For each response a LinFrameTriggering and a LinUnconditionalFrame shall be defined. Within a channel a LIN Frame shall be referenced by only one FrameTriggering. This allows a derivation of the identifier of a substituted Frame. The identifier is specified in FrameTriggering element. The Unconditional frames associated with an event triggered frame shall: • have equal length. • use the same checksum model (i.e. mixing LIN 1.x and LIN 2.x frames is not allowed). • reserve the first data field to its protected identifier (even if the associated unconditional frame is scheduled as a unconditional frame in the same or another schedule table). • be published by different slave nodes. • shall not be included directly in the same schedule table as the event triggered frame is scheduled.

Table 6.76: LinEventTriggeredFrame

6.7.2.2 LIN Schedule Table

The LinMaster uses one or more predefined scheduling tables to start the sending and receiving to the LIN bus. These scheduling tables contain at least the relative timing that defines the message sending.

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	***Datatype***	***Mul.***	***Kind***	***Note***
resumePosition	ResumePosition	0..1	attr	Defines, where a schedule table shall be proceeded in case if it has been interrupted by a run-once table or MRF/SRF.
runMode	RunMode	0..1	attr	The schedule table can be executed in two different modes.
tableEntry	ScheduleTableEntry	1..*	aggr	The scheduling table consists of table entries, which contain Frame slots.

Table 6.77: LinScheduleTable

Enumeration	**RunMode**
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication
Note	The schedule table can be executed in two different modes.
Literal	***Description***
RunContinu-ous	RUN_CONTINUOUS run mode
runOnce	RUN_ONCE run mode

Table 6.78: RunMode

Enumeration	**ResumePosition**
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication
Note	Defines, where a schedule table shall be proceeded in case if it has been interrupted by a run-once table or MRF/SRF.
Literal	***Description***
continueAtIt Position	Continue at IT Point.
startFrom Beginning	Start from the beginning

Table 6.79: ResumePosition

Class	**ScheduleTableEntry (abstract)**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	Table entry in a LinScheduleTable. Specifies what will be done in the frame slot.			
Base	ARObject			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
delay	TimeValue	1	attr	Relative delay between this tableEntry and the start of the successor in the schedule table in seconds.
introductio n	Documentation Block	0..1	aggr	This represents introductory documentation about the schedule table entry. **Tags:** xml.sequenceOffset=-10
positionInT able	Integer	1	attr	Relative position in the schedule table. The first entry index in the schedule table is 0.

Table 6.80: ScheduleTableEntry

Class	**ApplicationEntry**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	Schedule table entry for application messages.			
Base	ARObject,ScheduleTableEntry			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
frameTrigg ering	LinFrameTrigge ring	1	ref	Specifies the LinFrame that will be transmitted in this frame slot.

Table 6.81: ApplicationEntry

Class	FreeFormatEntry (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	FreeFormat transmits a fixed master request frame with the eight data bytes provided. This may for instance be used to issue user specific fixed frames.			
Base	ARObject,ScheduleTableEntry			
Attribute	Datatype	Mul.	Kind	Note
–	–	–	–	–

Table 6.82: FreeFormatEntry

Class	LinConfigurationEntry (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	A ScheduleTableEntry which contains LIN specific assignments.			
Base	ARObject,ScheduleTableEntry			
Attribute	Datatype	Mul.	Kind	Note
assignedController	LinSlave	1	ref	The LIN slaves controller who is target of this assignment.

Table 6.83: LinConfigurationEntry

6.7.2.3 Configuration Services

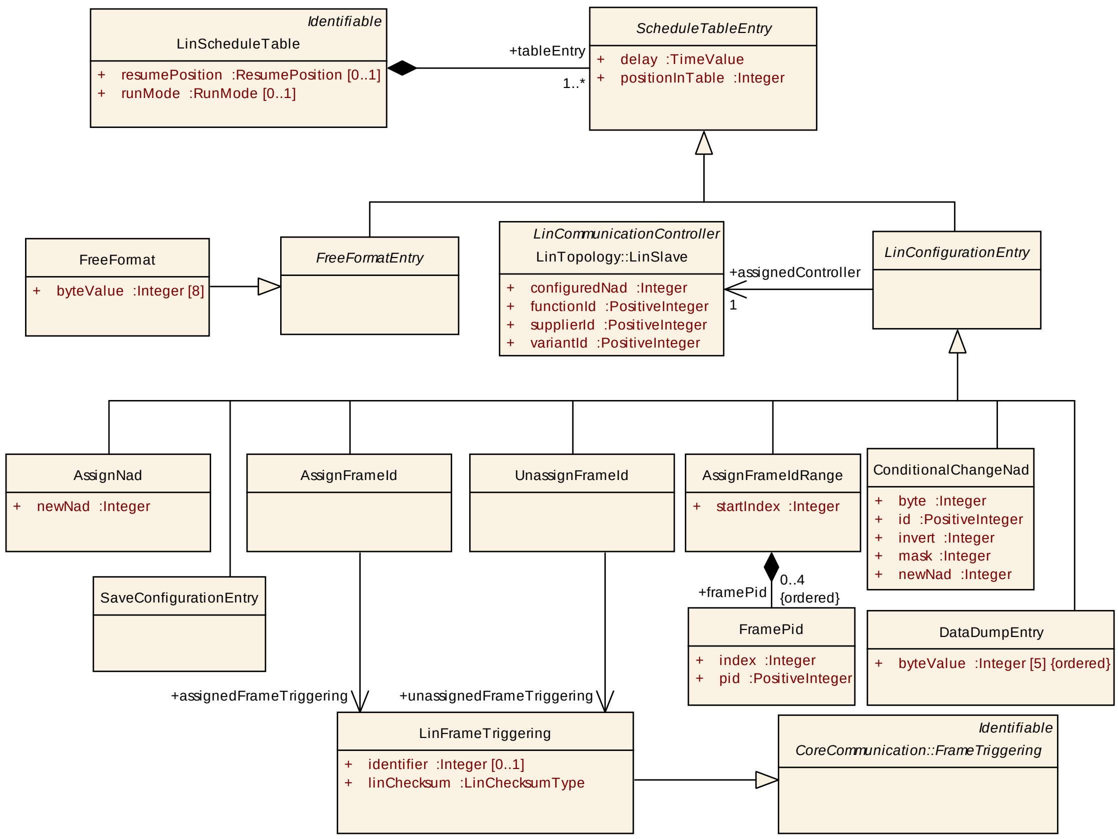

Figure 6.19: LIN Configuration Entries (Fibex4Lin:LinConfigurationEntries)

LIN only supports 64 identifiers. That creates the need for extending the address space. Hence the frames are identified by message ids from a much larger address space that is additionally separated by supplier ids. During runtime the master assigns a LinId to the frame. In case of identical parts within a cluster the initial node ID (oldNad) is used to differentiate such nodes.

To support that in System Template the `AssignFrameId` is introduced as a LIN specific extension. For the assignment a relation to the `LinSlave` is used. The `LinSlave` element is referenced by a `LinCommunicationConnector` element that contains a list of frames processed by the slave node. More details can be found in chapter 6.7.2.3.

Class	**AssignFrameId**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	Schedule entry for an Assign Frame Id master request.			
Base	ARObject,LinConfigurationEntry,ScheduleTableEntry			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
assignedFrameTriggering	LinFrameTriggering	1	ref	The frame whose identifier is set by this assignment.

Table 6.84: AssignFrameId

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	***Datatype***	***Mul.***	***Kind***	***Note***
unassignedFrameTriggering	LinFrameTriggering	1	ref	The frame whose identifier is reset by this assignment.

Table 6.85: UnassignFrameId

The Assign frame ID configuration service is replaced in LIN 2.1 by the Assign frame ID range configuration service. AssignFrameIdRange is used to set or disable Protected Identifiers up to four frames. For the assignment a relation to the LinSlave is used. The LinSlave element is referenced by a LinCommunicationConnector element that contains a list of frames processed by the slave node. More details can be found in chapter 6.7.2.3.

Class	**AssignFrameIdRange**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	AssignFrameIdRange generates an assign frame PID range request.			
Base	ARObject,LinConfigurationEntry,ScheduleTableEntry			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
framePid (ordered)	FramePid	0..4	aggr	Optional assignment of frame_PID values that are included in the request. The frame_PIDs are ordered.
startIndex	Integer	1	attr	The startIndex sets the index to the first frame to assign a PID.

Table 6.86: AssignFrameIdRange

Class	**FramePid**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	Frame_PIDs that are included in the request. The "pid" attribute describes the value and the "index" attribute the position of the frame_PID in the request.			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
index	Integer	1	attr	This attribute is used to order the frame_PIDs. The values of index shall be unique within one AssignFrameIdRange.
pid	PositiveInteger	1	attr	Frame_PID value.

Table 6.87: FramePid

Assign NAD is used to resolve conflicting NADs in LIN clusters built using off-the-shelves slave nodes or reused slave nodes. This request uses the initial NAD. The NAD used for the response shall be the same as in the request, i.e. the initial NAD.

Class	**AssignNad**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	Schedule entry for an Assign NAD master request.			
Base	ARObject,LinConfigurationEntry,ScheduleTableEntry			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
newNad	Integer	1	attr	The newly assigned NAD value.

Table 6.88: AssignNad

The conditional change NAD is used to detect unknown slave nodes in a cluster and to separate their NADs.

Class	**ConditionalChangeNad**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	Generates an conditional change NAD request. See LIN 2.1 protocol specification for more information.			
Base	ARObject,LinConfigurationEntry,ScheduleTableEntry			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
byte	Integer	1	attr	Byte Position of Data Byte that should be used for the bitwise XOR with Invert and the bitwise AND with Mask.
id	PositiveInteger	1	attr	Byte Position of Id.
invert	Integer	1	attr	Byte Position of Invert.
mask	Integer	1	attr	Byte Position of Mask.
newNad	Integer	1	attr	The newly assigned NAD value (Byte Position).

Table 6.89: ConditionalChangeNad

The Save Configuration service tells the slave node that the slave application shall save the current configuration.

Class	**SaveConfigurationEntry**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	This service is used to notify a slave node to store its configuration.			
Base	ARObject,LinConfigurationEntry,ScheduleTableEntry			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
–	–	–	–	–

Table 6.90: SaveConfigurationEntry

The Data Dump service is reserved for initial configuration of a slave node by the slave node supplier and the format of this message is supplier specific.

Class	**DataDumpEntry**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	This service is reserved for initial configuration of a slave node by the slave node supplier and the format of this message is supplier specific.			
Base	ARObject,LinConfigurationEntry,ScheduleTableEntry			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
byteValue (ordered)	Integer	5	attr	Supplier specific format.

Table 6.91: DataDumpEntry

With the FreeFormat a scheduling of fixed data content within a diagnostic frame is defined. For that specification `FreeFormat` is introduced.

Class	**FreeFormat**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinCommunication			
Note	Representing freely defined data.			
Base	ARObject,FreeFormatEntry,ScheduleTableEntry			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
byteValue	Integer	8	attr	The integer Value of a freely defined data byte.

Table 6.92: FreeFormat

In order to be consistent with the rest of the communication configuration, it is required that the diagnostic LIN Frames (Master Request Frame, Slave Response Frame) are explicitly modeled as `Frame` elements. `LinFrameTriggering`s dealing with diagnostic Frames thus reference this diagnostic frames.

6.7.3 CAN specific description

This chapter describes additions to the CAN definition of FrameTriggerings.

[TPS_SYST_01130] Communication over CAN ⌈ The System Template supports the description of communication over CAN. ⌋*(RS_SYST_00021)*

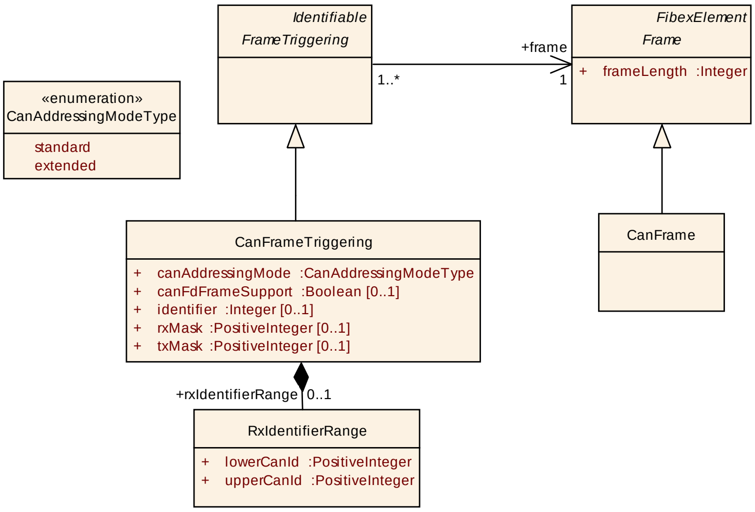

Figure 6.20: CanFrameTriggering (Fibex4Can:CanCommunication)

Class	**CanFrame**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanCommunication			
Note	CAN specific Frame element. This element shall also be used for TTCan. **Tags:** atp.recommendedPackage=Frames			
Base	ARObject,CollectableElement,FibexElement,Frame,Identifiable,Multilanguage Referrable,PackageableElement,Referrable			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
–	–	–	–	–

Table 6.93: CanFrame

Class	**CanFrameTriggering**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanCommunication			
Note	CAN specific attributes to the FrameTriggering			
Base	ARObject,FrameTriggering,Identifiable,MultilanguageReferrable,Referrable			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
absolutely Scheduled Timing	TtcanAbsolutely ScheduledTimin g	*	aggr	Each frame in TTCAN is identified by its slot id and communication cycle. A description is provided by the usage of AbsolutelyScheduledTiming.
canAddres singMode	CanAddressing ModeType	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.

Attribute	Datatype	Mul.	Kind	Note
canFdFrameSupport	Boolean	0..1	attr	This attribute describes whether the CAN FD option is activated or not. If this attribute is TRUE this frame can be sent/received as CAN FD frame. Otherwise it has to be CAN 2.0 compliant.
identifier	Integer	0..1	attr	To describe a frames identifier on the communication system, usually with a fixed identifierValue. In a complete system description this attribute is mandatory. In an Ecu Extract for the sender of the frame the identifier shall also be provided. In an Ecu Extract for the receiver the identifier attribute shall be ignored if rxIdentifierRange is defined.
rxIdentifierRange	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 6.94: CanFrameTriggering

Enumeration	**CanAddressingModeType**
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanCommunication
Note	Indicates whether standard or extended CAN identifiers are used
Literal	*Description*
extended	Extended 29-bit-identifiers are used (CAN 2.0B)
standard	Standard 11-bit-identifiers are used (CAN 2.0A)

Table 6.95: CanAddressingModeType

Class	**RxIdentifierRange**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanCommunication			
Note	Optional definition of a CanId range to reduce the effort of specifying every possible FrameTriggering within the defined Id range during reception. All frames received within a range are mapped to the same Pdu that is passed to a upper layer module (e.g. Nm, CDD, PduR). This range is redundant to the nmRangeConfig attributes of "CanNmNode". For backward compatibility reasons this redundancy shall be preserved and both shall be defined.			
Base	ARObject			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
lowerCanId	PositiveInteger	1	attr	This attribute can be used together with the upperCanId attribute to define a range of CanIds.
upperCanId	PositiveInteger	1	attr	This attribute can be used together with the lowerCanId attribute to define a range of CanIds.

Table 6.96: RxIdentifierRange

6.7.3.1 SAE J1939 Protocol specific description

J1939 is a protocol and application layer standard of the SAE (Society of Automotive Engineers) based on the CAN technology. It defines parameters uniquely identified by the SPN (Suspect Parameter Number). These are mapped to parameter groups that are uniquely identified by a PGN (Parameter Group Number). Parameters are simply handled as SystemSignals which have a name derived from the name of the SPNs. A Parameter Group (PG) corresponds to an IPdu.

J1939 uses extended 29 bit CAN identifiers to encode a priority, the source address of the frame, and a frame ID which is based on the PGN (Parameter Group Number) and may contain the destination address.

J1939 supports IPdus with more than 8 bytes, and IPdus with variable length that may exceed 8 bytes. As soon as an IPdu has more than 8 bytes, it does not fit in a single CAN frame and a transport protocol must be used. Variable length IPdus will always be handled by the J1939 TP, regardless of the actual length. The J1939 Transport Protocol is described in chapter 6.8.6.

[TPS_SYST_01132] Communication over SAE J1939 ⌈ The System Template supports the description of communication over SAE J1939. ⌋*(RS_SYST_00038)*

6.7.4 TTCAN specific description

This chapter describes additions to the TTCAN definition of `FrameTriggering`s.

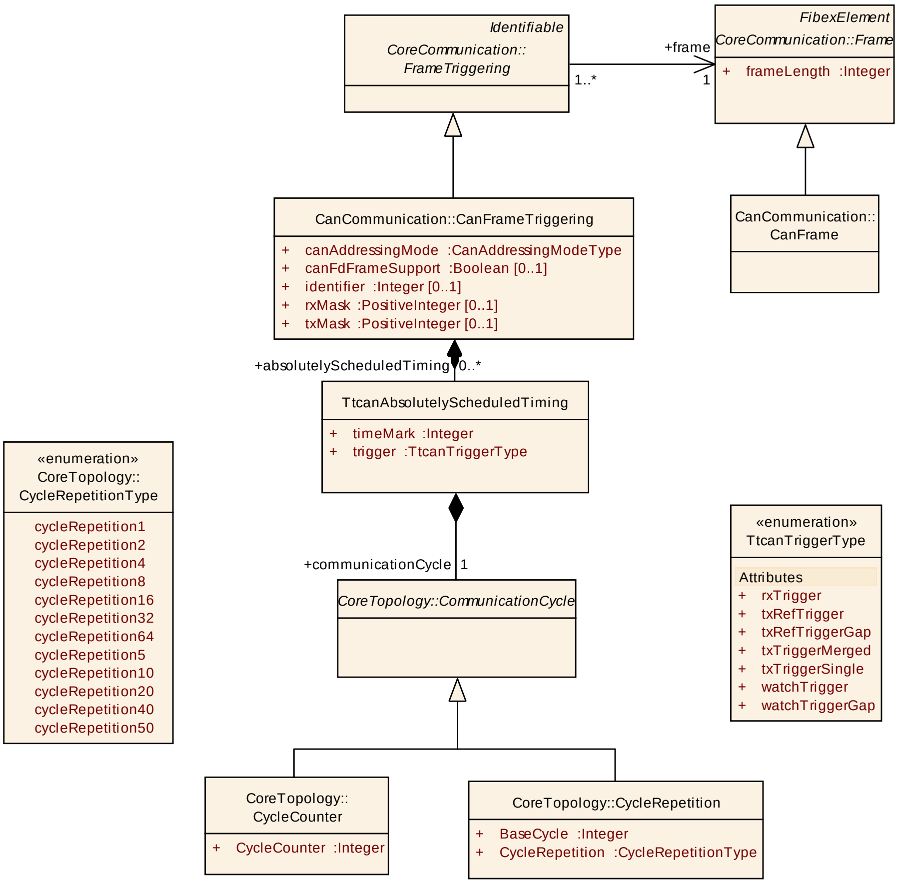

Figure 6.21: TtcanAbsolutelyScheduledTiming (Fibex4Ttcan:TtcanCommunication)

Class	**TtcanAbsolutelyScheduledTiming**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ttcan::Ttcan Communication			
Note	Each frame in TTCAN is identified by its slot id and communication cycle. A description is provided by the usage of AbsolutelyScheduledTiming. A frame can be sent multiple times within one communication cycle. For describing this case multiple AbsolutelyScheduledTimings have to be used. The main use case would be that a frame is sent twice within one communication cycle.			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
communicationCycle	CommunicationCycle	1	aggr	The communication cycle where the frame is sent.
timeMark	Integer	1	attr	Where FlexRay counts the slots in the static segment, TTCAN requires explicit Tx and Rx time marks.
trigger	TtcanTriggerType	1	attr	Trigger type for this time window.

Table 6.97: TtcanAbsolutelyScheduledTiming

Enumeration	**TtcanTriggerType**
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ttcan::Ttcan Communication
Note	This type lists all trigger types for a time window.
Literal	**Description**
rxTrigger	Check for message reception
txRefTrigger	Send reference message in periodic case
txRefTriggerGap	Send reference message in event-synchronised case
txTriggerMerged	Send message in a merged arbitration window
txTriggerSingle	Send message in an exclusive time window
watchTrigger	Check for missing reference message in periodic case
watchTriggerGap	Check for missing reference message in event-synchronised case

Table 6.98: TtcanTriggerType

6.7.5 Ethernet specific description

[TPS_SYST_01131] TCP/IP and UDP/IP communication over Ethernet ⌈ The System Template supports the description of TCP/IP and UDP/IP communication over Ethernet. ⌋*(RS_SYST_00039)*

This section specifies the information of the AUTOSAR Basic Software modules Socket Adaptor (SoAd), Service Discovery and Tcp/Ip that is common for several `EcuInstance`s and therefore is part of the System Configuration Description.

The main purpose of the SoAd module is to create an interface between the PDU Router and a socket based TCP/IP stack. The AUTOSAR Service Discovery module offers functionality to detect and announce available services within the vehicle network.

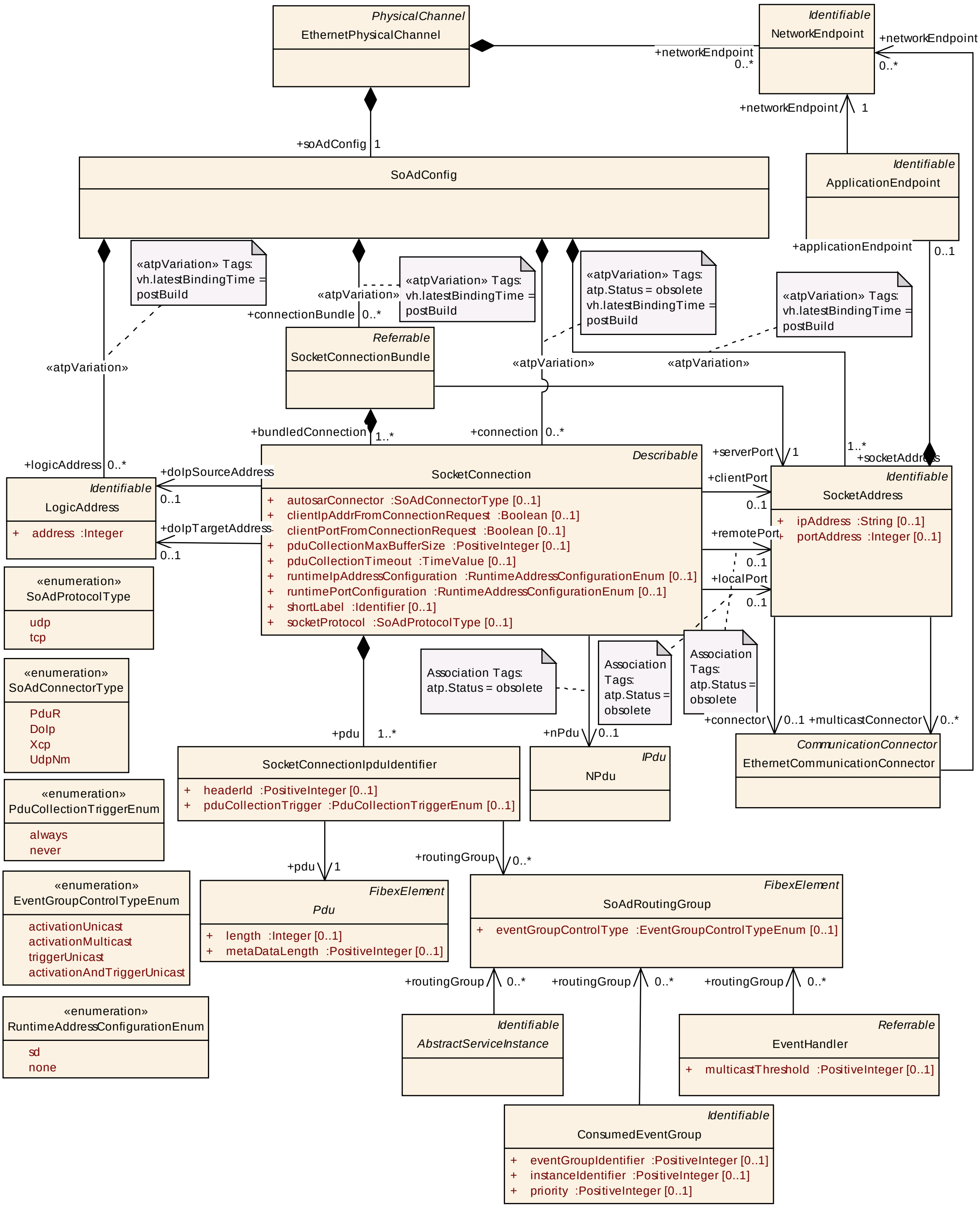

Figure 6.22: Ethernet Communication (Fibex4Ethernet:EthernetCommunication)

The SoAd serves as a (De)Multiplexer between different Pdu sources/suppliers and the TCP/IP stack. The `SocketConnection` maps TCP/UDP Ports (configured by the `ApplicationEndpoint`) as well as IP addresses (configured by the `NetworkEndpoint`) to the `IPdu` and adds this information during transmit. On receive it needs to reverse this process and create the `IPdu` from the TCP/IP information received.

The SocketConnectionBundle groups SocketConnections and describes properties like serverPorts which are common for all SocketConnections in the SocketConnectionBundle.

[TPS_SYST_01091] Definition of SoAdConfig ⌈ The SoAdConfig in the System Template is defined per EthernetPhysicalChannel which represents a VLAN. ⌋*(RS_SYST_00039)*

[TPS_SYST_01092] Transmission of multiple Pdus over the same SocketConnection ⌈ If multiple Pdus are transmitted over the same SocketConnection a headerId information shall be used to distinguish between the different Pdus. ⌋*(RS_SYST_00039)*

[TPS_SYST_01093] Activation/Deactivation of SoAdRoutingGroups ⌈ The routing of Pdus to and from a socket may be activated or deactivated with a SoAdRoutingGroup depending on the availability of services, EventHandlers or ConsumedEventGroups that send or receive the data. ⌋*(RS_SYST_00039)*

The Routing Group Activation Table is controlled by the Service Discovery module.

[TPS_SYST_02002] SoAdRoutingGroup for Services with Methods ⌈ For Services that contain Methods a MethodActivationRoutingGroup shall be created that is referenced by the ProvidedServiceInstance and by all clients in form of ConsumedServiceInstances. ⌋

[TPS_SYST_02003] SoAdRoutingGroups for Services with event groups ⌈ For event groups of a Service an EventGroupActivationGroup (with eventGroupControlType set to activationUnicast or activationMulticast) shall be created that is referenced by the EventHandler and by all ConsumedEventGroups that are subscribed to this event group. ⌋

[TPS_SYST_02004] SoAdRoutingGroups for Services with event groups that contain triggered events ⌈ A TriggerRoutingGroup (with eventGroupControlType set to triggerUnicast) shall be created for event groups that contain triggered events in addition to an EventGroupActivationGroup. Such a TriggerRoutingGroup shall be referenced by the EventHandler that provides the triggered events. ⌋

The Methods and Events that are provided or consumed by an EcuInstance are described by the ApplicationEndpoint.

Class	**SoAdConfig**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::Ethernet Communication			
Note	SoAd Configuration for one specific Physical Channel.			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**

Attribute	Datatype	Mul.	Kind	Note
connection	SocketConnection	*	aggr	This aggregation is obsolete and will be removed in the future. The connectionGroup aggregation with bundledConnections shall be used instead. Old description: Collection of socket connections. **Stereotypes:** atpVariation **Tags:** atp.Status=obsolete vh.latestBindingTime=postBuild
connectionBundle	SocketConnectionBundle	*	aggr	Collection of SocketConnectionBundles. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
logicAddress	LogicAddress	*	aggr	Collection of DoIp Addresses. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
socketAddress	SocketAddress	1..*	aggr	Collection of SoAdAddresses. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild

Table 6.99: SoAdConfig

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	*Datatype*	*Mul.*	*Kind*	*Note*
bundledConnection	SocketConnection	1..*	aggr	Collection of SocketConnections in the connectionGroup.
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.

Table 6.100: SocketConnectionBundle

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	*Datatype*	*Mul.*	*Kind*	*Note*

Attribute	Datatype	Mul.	Kind	Note
autosarConnector	SoAdConnectorType	0..1	attr	This attribute is deprecated and will be removed in future. **Tags:** atp.Status=obsolete
clientIpAddrFromConnectionRequest	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.
clientPortFromConnectionRequest	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.
doIpSourceAddress	LogicAddress	0..1	ref	The logical DoIP address of the source entity. This optional reference shall only be used for DoIP (Diagnosis over IP).
doIpTargetAddress	LogicAddress	0..1	ref	The logical DoIP address of the target entity. This optional reference shall only be used for DoIP (Diagnosis over IP).
localPort	SocketAddress	0..1	ref	This reference is obsolete and will be removed in the future. The serverPort reference in SocketConnectionBundle shall be used instead. Old description: Local Port for TCP/UDP connection. **Tags:** atp.Status=obsolete
nPdu	NPdu	0..1	ref	Reference to data packets that are transmitted over Ethernet. Each data packet can contain multiple IPdus. Please note that this reference is deprecated. **Tags:** atp.Status=obsolete
pdu	SocketConnectionIpduIdentifier	1..*	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.
pduCollectionMaxBufferSize	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.

Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
pduCollectionTimeout	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.
remotePort	SocketAddress	0..1	ref	This reference is obsolete and will be removed in the future. The clientPort reference shall be used instead. Old description: Remote Port for TCP/UDP connection. May be different for each Frame or use the same remote port. In second case headerId attribute needs to be considered. **Tags:** atp.Status=obsolete
runtimeIpAddressConfiguration	RuntimeAddressConfigurationEnum	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.
runtimePortConfiguration	RuntimeAddressConfigurationEnum	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 SocketConnection. If this attribute is set to none the client uses the statically configured Port information.
shortLabel	Identifier	0..1	ref	This attribute specifies an identifying shortName for the SocketConnection. It shall be unique within its context.
socketProtocol	SoAdProtocolType	0..1	attr	This attribute is deprecated and will be removed in future. **Tags:** atp.Status=obsolete

Table 6.101: SocketConnection

Figure 6.23 shows an example with a `SocketConnectionBundle` that contains three `SocketConnection`s. `Client1Connection` and `Client2Connection` are pointing to static unicast addresses. `MulticastConnection` is pointing to a multicast IP Address.

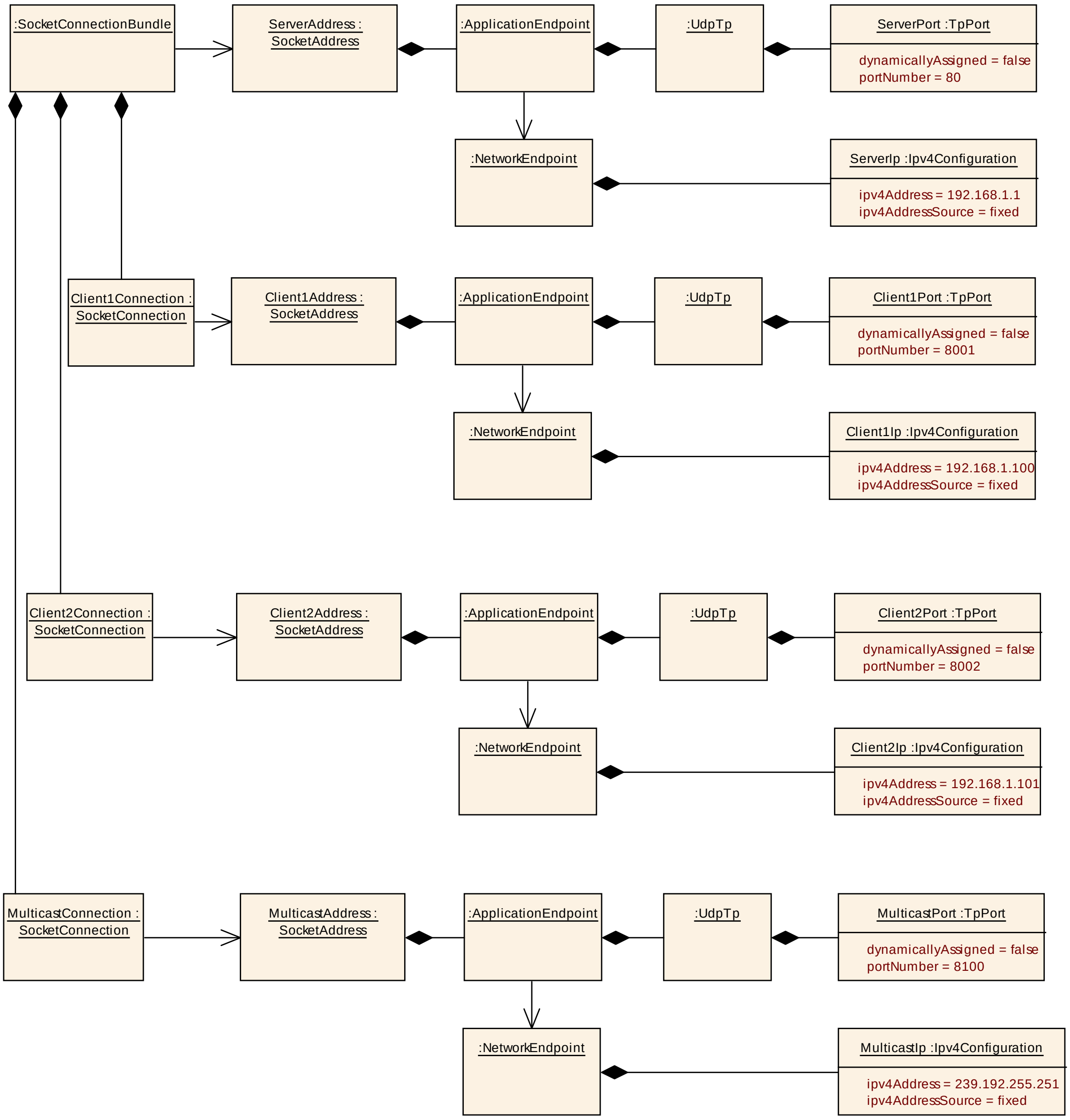

Figure 6.23: Base Addressing Example

Enumeration	**RuntimeAddressConfigurationEnum**
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::Ethernet Communication
Note	This enumeration defines the protocol to be used to obtain the address information.
Literal	***Description***
none	Static configuration is used to obtain the address information.
sd	AUTOSAR Service Discovery is used to obtain the address information.

Table 6.102: RuntimeAddressConfigurationEnum

Class	SocketConnectionIpduIdentifier			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::Ethernet Communication			
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	Datatype	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.
pdu	Pdu	1	ref	Reference to an IPdu that is mapped to a socket connection.
pduCollectionTrigger	PduCollectionTriggerEnum	0..1	attr	Defines whether the referenced Pdu contributes to the triggering of the socket transmission if Pdu collection is enabled for this socket.
routingGroup	SoAdRoutingGroup	*	ref	Reference to RoutingGroups that can be enabled or disabled.

Table 6.103: SocketConnectionIpduIdentifier

Enumeration	PduCollectionTriggerEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::Ethernet Communication
Note	Defines whether a Pdu contributes to the triggering of the socket transmission if Pdu collection is enabled for the socket.
Literal	Description
always	Pdu will trigger the transmission of the socket buffer.
never	Pdu will be buffered and will not trigger the transmission of the socket buffer.

Table 6.104: PduCollectionTriggerEnum

Class	SocketAddress			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::Ethernet Communication			
Note	SocketAddress contains the portAddress (in the ApplicationEndpoint) and the ipAddress (in the NetworkEndpoint that is referenced by the ApplicationEndpoint).			
Base	ARObject,Identifiable,MultilanguageReferrable,Referrable			
Attribute	Datatype	Mul.	Kind	Note
applicationEndpoint	ApplicationEndpoint	0..1	aggr	Application addressing

Attribute	Datatype	Mul.	Kind	Note
connector	EthernetCommunicationConnector	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.
ipAddress	String	0..1	attr	This attribute is deprecated and will be removed in future. It is replaced by the aggregated NetworkEndpoint. **Tags:** atp.Status=obsolete
multicastConnector	EthernetCommunicationConnector	*	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.
portAddress	Integer	0..1	attr	This attribute is deprecated and will be removed in future. It is replaced by the aggregated ApplicationEndpoint. **Tags:** atp.Status=obsolete

Table 6.105: SocketAddress

Class	**SoAdRoutingGroup**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication			
Note	Routing of Pdus in the SoAd can be activated or deactivated. The ShortName of this element shall contain the RoutingGroupId. **Tags:** atp.recommendedPackage=SoAdRoutingGroups			
Base	ARObject,CollectableElement,FibexElement,Identifiable,MultilanguageReferrable,PackageableElement,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***

Attribute	Datatype	Mul.	Kind	Note
eventGroupControlType	EventGroupControlTypeEnum	0..1	attr	This attribute defines the type of a RoutingGroup. There are RoutingGroups that activate the data path for unicast or multicast events of an event group. And there are RoutingGroups that activate the data path for initial events that are triggered, namely events that are sent out on the server side after a client got subscribed. Please note that this attribute is only valid for event communication (Sender Receiver communication) and shall be omitted in MethodActivationRoutingGroups.

Table 6.106: SoAdRoutingGroup

Enumeration	EventGroupControlTypeEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication
Note	Types of a RoutingGroups for the event communication.
Literal	***Description***
activationAndTriggerUnicast	Activate the data path for unicast events and triggered unicast events that are sent out after a client got subscribed.
activationMulticast	Activate the data path for multicast events of an EventGroup.
activationUnicast	Activate the data path for unicast events of an EventGroup.
triggerUnicast	Activate the data path for triggered unicast events that are sent out after a client got subscribed.

Table 6.107: EventGroupControlTypeEnum

The element NPdu that is optionally referenced by the SocketConnection is used to describe the datagram that is transmitted from the Tcp/Ip to the Ethernet Interface. This datagram can be mapped into an EthernetFrame with the PduToFrameMapping element. Please note that with the introduction of the TcpIp Bsw module the description of EthernetFrames is no longer necessary for configuration of the AUTOSAR Ethernet Stack. Therefore the nPdu reference in the SocketConnection is set to deprecated. This means that Pdus that are transmitted over a SocketConnection are not mapped into Frames by a PduToFrameMapping element.

Class	EthernetFrame			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::Ethernet Communication			
Note	Ethernet specific attributes to the Frame **Tags:** atp.recommendedPackage=Frames			
Base	ARObject,CollectableElement,FibexElement,Frame,Identifiable,Multilanguage Referrable,PackageableElement,Referrable			
Attribute	Datatype	Mul.	Kind	Note
–	–	–	–	–

Table 6.108: EthernetFrame

Class	EthernetFrameTriggering			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::Ethernet Communication			
Note	Ethernet specific Frame element.			
Base	ARObject,FrameTriggering,Identifiable,MultilanguageReferrable,Referrable			
Attribute	Datatype	Mul.	Kind	Note
–	–	–	–	–

Table 6.109: EthernetFrameTriggering

6.7.5.1 Ethernet Addressing examples

This chapter describes how the attributes clientPortFromConnectionRequest, runtimePortConfiguration , clientIpAddressFromConnectionRequest and runtimeIpAddressConfiguration are used. Please note that the System with category SYSTEM_EXTRACT is a subset of a System with category SYSTEM_DESCRIPTION and that no data shall be altered during the creation of the System Extract. This is also true for the values of attributes that are described in this chapter.

For more details about the System with category SYSTEM_EXTRACT see chapter 9.

[TPS_SYST_02007] Usage of SocketConnection attributes in the unicast server view ⌈ In the unicast server view the following rules apply:

- If the clientPortFromConnectionRequest is set to false the Server obtains the client Port from the static configuration.
- If the clientPortFromConnectionRequest is set to true the Server obtains the client Port from the information contained in the connection request at run-time.
- If the clientIpAddressFromConnectionRequest is set to false the Server obtains the client IpAddress from the static configuration.

- If the clientIpAddressFromConnectionRequest is set to true the Server obtains the client IpAddress from the information contained in the connection request at runtime.

⌋

[TPS_SYST_02008] Usage of SocketConnection attributes in the unicast client view ⌈
In the unicast client view the following rules apply:

- If the runtimePortConfiguration attribute is set to none the client obtains the server port from the static configuration.
- If the runtimePortConfiguration attribute is set to other than none the client uses the respective protocol to retrieve the port information at runtime.
- If the runtimeIpAddressConfiguration attribute is set to none the client obtains the server IpAddress from the static configuration.
- If the runtimeIpAddressConfiguration attribute is set to other than none the client uses the respective protocol to retrieve the IpAddress information at runtime.

⌋

Figure 6.24 shows two unicast use cases. In the first one (Client1Connection) both client and server configure port number and IP address of the remote partner at configuration time. In the second use case (Client2Connection) both client and server configure port number and IP address of the remote partner at runtime. The server obtains the clients IP address and port number when a connection request from any client arrives. The client obtains the server port number and IP address at runtime via the Sd service.

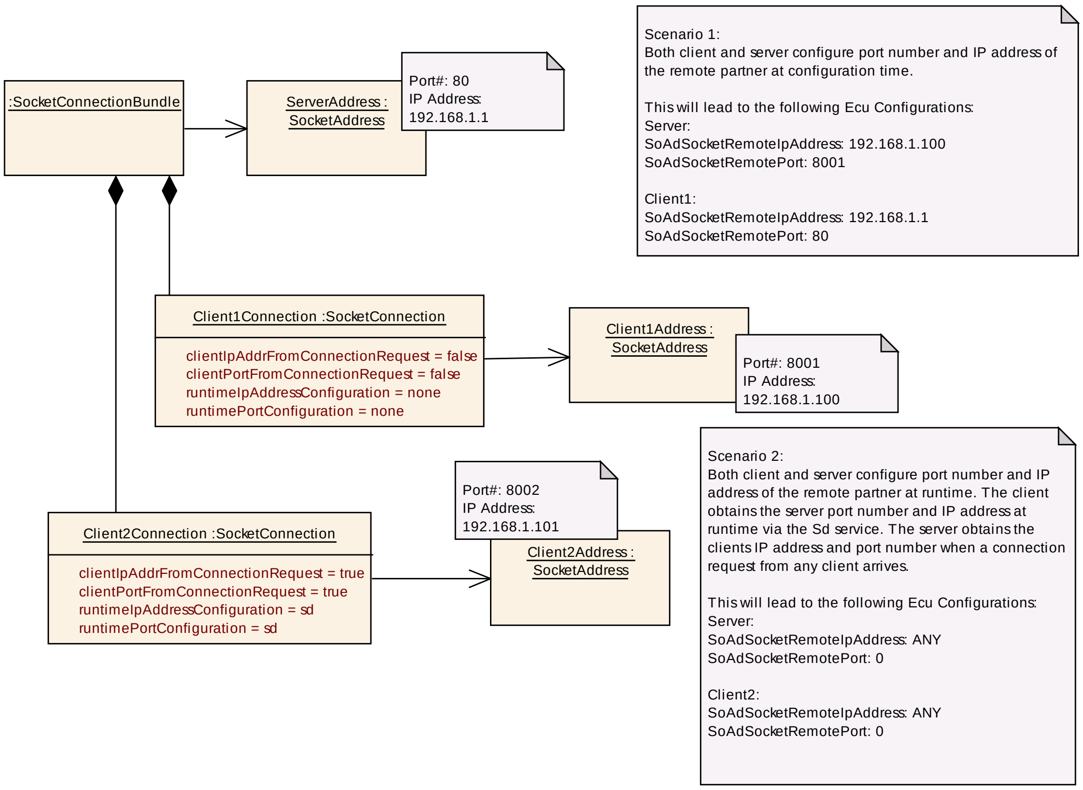

Figure 6.24: Unicast use cases 1 and 2

Figure 6.25 shows two additional unicast use cases. In the first one the server obtains the clients IP address and port number when a connection request from any client arrives. The client configures port number and IP address of the server at configuration time. In use case 4 the server obtains the clients IP address and port number at configuration time. The client obtains the server port number and IP address at runtime via the Sd service.

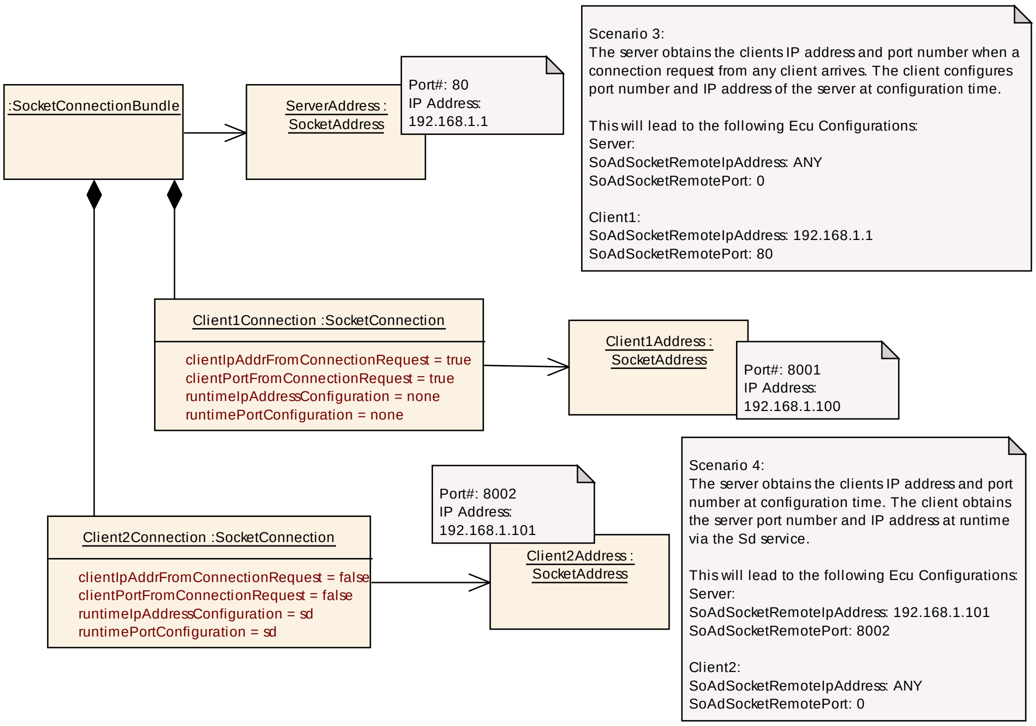

Figure 6.25: Unicast use cases 3 and 4

[TPS_SYST_02009] Usage of SocketConnection attributes in the multicast server view ⌈ In the multicast server view the following rules apply:

- The clientPortFromConnectionRequest attribute is ignored. The server always configures the remote port information based on the associated ApplicationEndpoint of the clientPort.
- The clientIpAddressFromConnectionRequest attribute is ignored. The server always configures the remote IpAddress information based on the associated networkEndpoint of the clientPort.

⌋

[TPS_SYST_02010] Usage of SocketConnection attributes in the multicast client view ⌈ In the multicast client view the following rules apply:

- If the runtimePortConfiguration attribute is set to none the client obtains its own port information from the static configuration based on the associated ApplicationEndpoint of the clientPort.
- If the runtimePortConfiguration attribute is set to other than none the client uses the respective protocol to retrieve its own port information at runtime.
- If the runtimeIpAddressConfiguration attribute is set to none the client obtains its own port information from the static configuration based on the associated networkEndpoint of the clientPort.

- If the runtimeIpAddressConfiguration attribute is set to other than none the client uses the respective protocol to retrieve its own port information at run-time.

⌋

Figure 6.26 shows two multicast use cases. In the first one the Multicast address is known at configuration time: The server configures the remote port number and IP address statically. Every addressed client adds a local address entry to its TcpIp configuration and configures the local port number statically.

The second use case shows a dynamic assignment of a Multicast address: The Server announces the Multicast address and port number it uses via Sd. Interested clients reconfigure the appropriate entries in their local configuration tables at runtime.

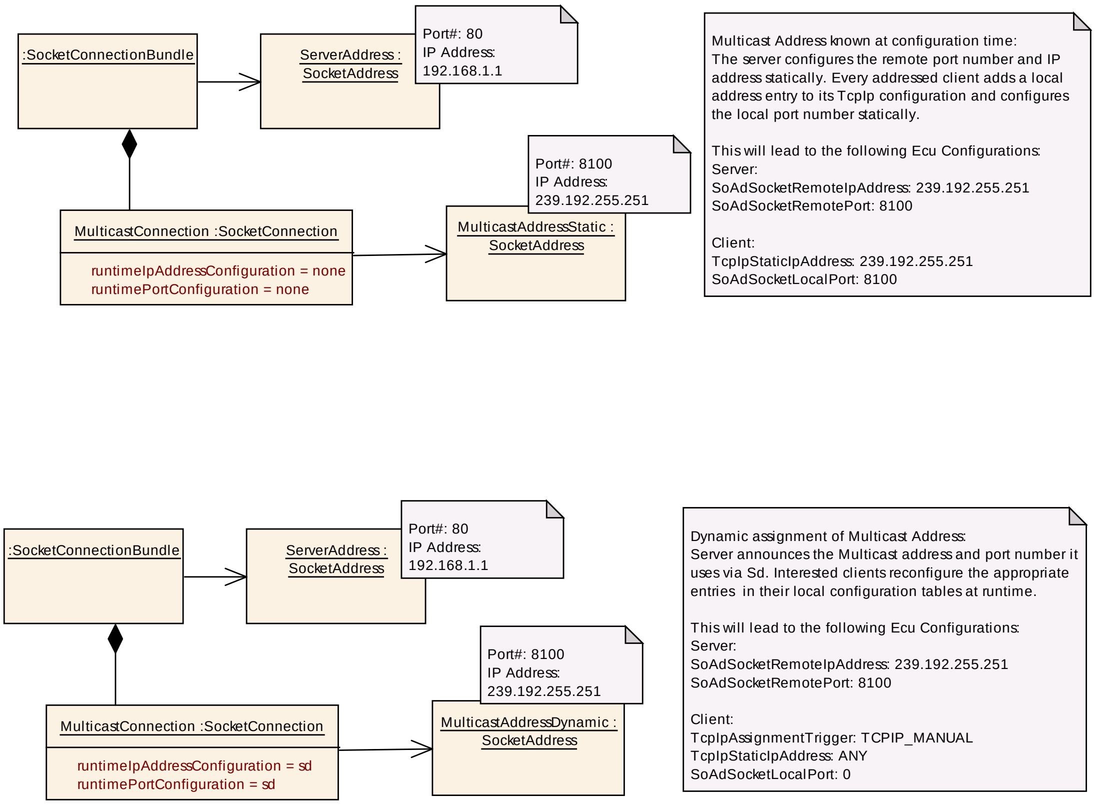

Figure 6.26: Multicast Example

6.7.5.2 Network Endpoint

The NetworkEndpoint defines the network addressing. The network endpoint may have a priority and a FQDN (Fully Qualified Domain Name) that is used for the Service Discovery (e.g. some.example.host.). Each NetworkEndpoint is related to exactly one CommunicationController via the SocketAddress and the reference to the EthernetCommunicationConnector.

[TPS_SYST_01090] valid NetworkEndpoint ⌈ To build a valid NetworkEndpoint a MacMulticastConfiguration with a reference to a MacMulticastGroup or an IP configuration (Ipv4Configuration or Ipv6Configuration) needs to be defined. ⌋*(RS_SYST_00039)*

The reference to the MacMulticastGroup is needed for the mapping of IP multicast to MAC multicast.

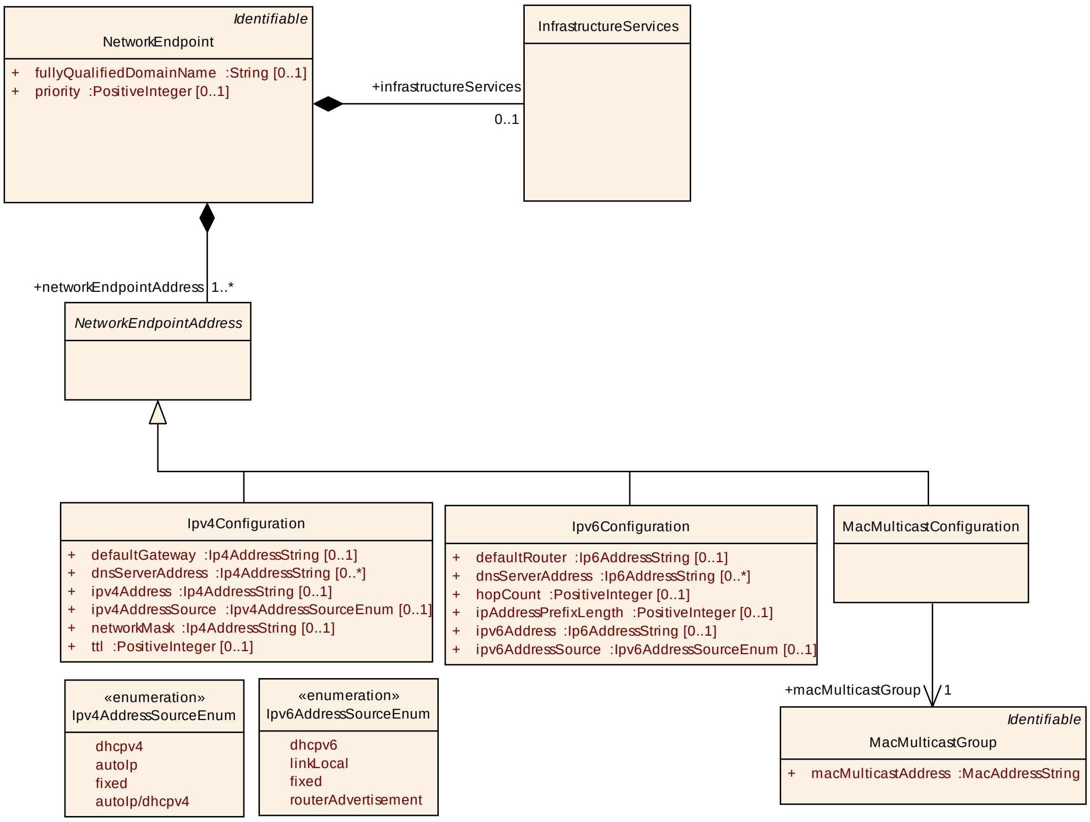

Figure 6.27: Network Endpoint

[TPS_SYST_01088] NetworkEndpoint priority ⌈ The priority at the NetworkEndpoint shall be used as Ethernet Header information together with the vlanIdentifier. If defined the priority overwrites the defaultPriority that is defined in the VlanMembership. The priority zero represents the highest priority. ⌋*(RS_SYST_00039)*

Class	**NetworkEndpoint**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::Ethernet Topology			
Note	The network endpoint defines the network addressing (e.g. IP-Address or MAC multicast address).			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
fullyQualifiedDomainName	String	0..1	attr	Defines the fully qualified domain name (FQDN) e.g. some.example.host.

Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
infrastructureServices	InfrastructureServices	0..1	aggr	Defines the network infrastructure services provided or consumed.
networkEndpointAddress	NetworkEndpointAddress	1..*	aggr	Definition of a Network Address. **Tags:** xml.namePlural=NETWORK-ENDPOINT-ADDRESSES
priority	PositiveInteger	0..1	attr	Priority of this Network-Endpoint. Zero representing the highest Priority.

Table 6.110: NetworkEndpoint

Class	**NetworkEndpointAddress (abstract)**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	To build a valid network endpoint address there has to be either one MAC multicast group reference or an ipv4 configuration or an ipv6 configuration.			
Base	ARObject			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 6.111: NetworkEndpointAddress

Class	**Ipv4Configuration**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Internet Protocol version 4 (IPv4) configuration.			
Base	ARObject,NetworkEndpointAddress			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
defaultGateway	Ip4AddressString	0..1	attr	IP address of the default gateway.
dnsServerAddress	Ip4AddressString	*	attr	IP addresses of preconfigured DNS servers. **Tags:** xml.namePlural=DNS-SERVER-ADDRESSES
ipv4Address	Ip4AddressString	0..1	attr	IPv4 Address. Notation: 255.255.255.255. The IP Address shall be declared in case the ipv4AddressSource is FIXED and thus no auto-configuration mechanism is used.
ipv4AddressSource	Ipv4AddressSourceEnum	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 6.112: Ipv4Configuration

Enumeration	**Ipv4AddressSourceEnum**
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology
Note	Defines how the node obtains its IPv4-Address.
Literal	***Description***
autoIp	AutoIP is used to dynamically assign IP addresses at device startup.
autoIpdhcpv4	The IpAddress is declared via AutoIp or dhcp.
dhcpv4	DHCP is a service for the automatic IP configuration of a client.
fixed	The IP Address shall be declared manually.

Table 6.113: Ipv4AddressSourceEnum

Class	**Ipv6Configuration**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Internet Protocol version 6 (IPv6) configuration.			
Base	ARObject,NetworkEndpointAddress			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
defaultRouter	Ip6AddressString	0..1	attr	IP address of the default router.
dnsServerAddress	Ip6AddressString	*	attr	IP addresses of pre configured DNS servers. **Tags:** xml.namePlural=DNS-SERVER-ADDRESSES
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)
ipAddressPrefixLength	PositiveInteger	0..1	attr	IPv6 prefix length defines the part of the IPv6 address that is the network prefix.
ipv6Address	Ip6AddressString	0..1	attr	IPv6 Address. Notation: FFFF:...:FFFF. The IP Address shall be declared in case the ipv6AddressSource is FIXED and thus no auto-configuration mechanism is used.
ipv6AddressSource	Ipv6AddressSourceEnum	0..1	attr	Defines how the node obtains its IP address.

Table 6.114: Ipv6Configuration

Enumeration	**Ipv6AddressSourceEnum**
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology
Note	Defines how the node obtains its IPv6-Address.
Literal	*Description*
dhcpv6	DHCP is a service for the automatic IP configuration of a client.
fixed	The IP Address shall be declared manually.
linkLocal	LinkLocal is intended only for communications within the segment of a local network (a link) or a point-to-point connection that a host is connected to.
routerAdvertisement	IPv6 Stateless Autoconfiguration.

Table 6.115: Ipv6AddressSourceEnum

Class	**MacMulticastConfiguration**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	References a per cluster globally defined MAC-Multicast-Group.			
Base	ARObject,NetworkEndpointAddress			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
macMulticastGroup	MacMulticastGroup	1	ref	Reference to a macMulticastGroup.

Table 6.116: MacMulticastConfiguration

In addition infrastructure services may be provided or consumed by the `NetworkEndpoint`s.

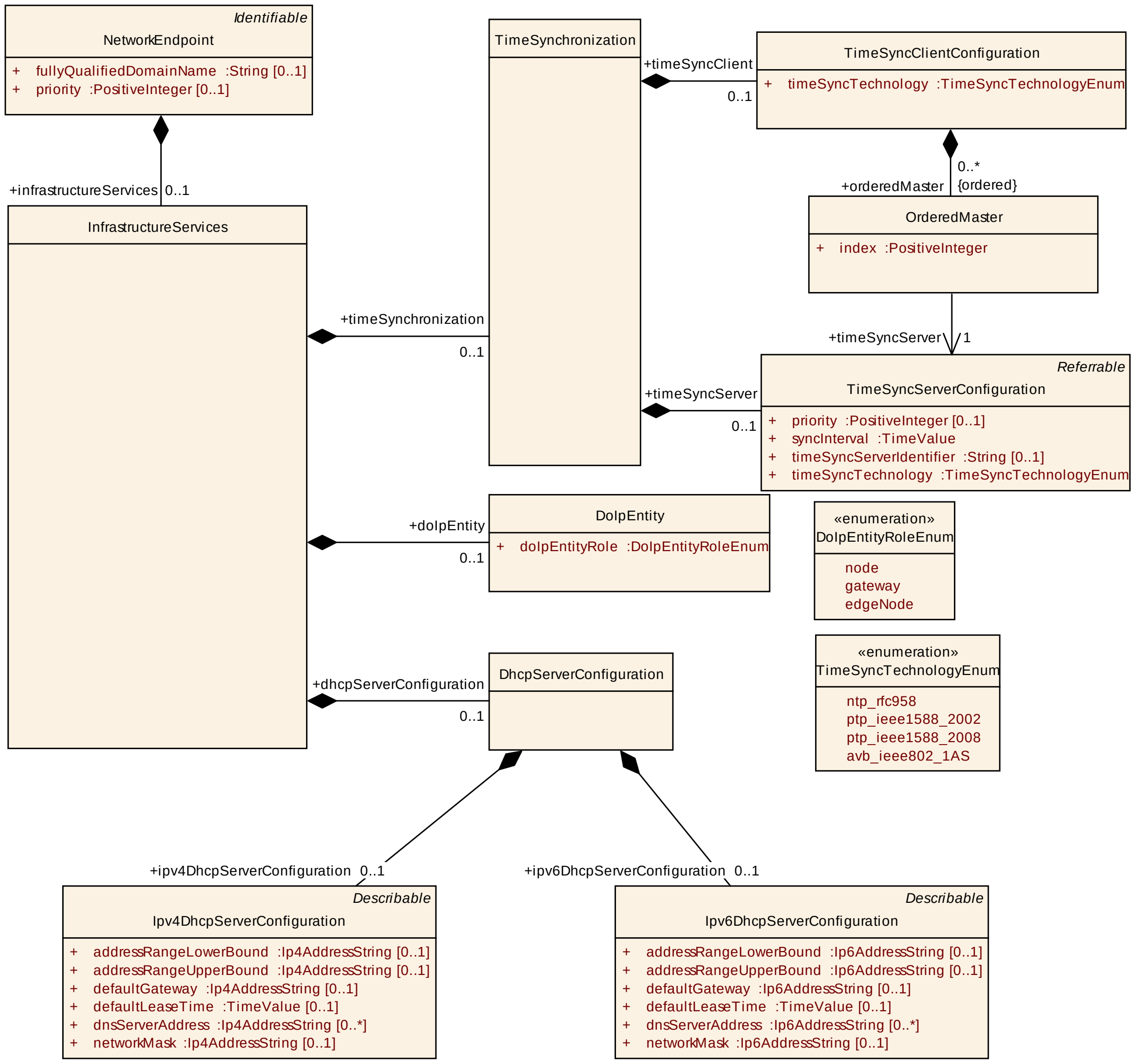

Figure 6.28: Network Endpoint Infrastructure Services

Class	**InfrastructureServices**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Defines the network infrastructure services provided or consumed.			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
dhcpServerConfiguration	DhcpServerConfiguration	0..1	aggr	Defines the configuration of DHCP servers that are running on the network endpoint.
doIpEntity	DoIpEntity	0..1	aggr	Defines whether a infrastructure service that runs on the network endpoint is a DoIP-Entity.
timeSynchronization	TimeSynchronization	0..1	aggr	Defines the servers / clients in a time synchronised network.

Table 6.117: InfrastructureServices

One of these services is a DHCP Server. The DHCP Server offers a service for the automatic IP-configuration of a client. This service is consumed by all DHCP clients in a subnet which have set the address source attribute in the IP Configuration to dhcpv4 or dhcpv6.

Class	**DhcpServerConfiguration**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::Ethernet Topology			
Note	Defines the configuration of DHCP servers that are running on the network endpoint. It is possible that an Ipv4DhcpServer and an Ipv6DhcpServer run on the same Ecu.			
Base	ARObject			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
ipv4DhcpServerConfiguration	Ipv4DhcpServerConfiguration	0..1	aggr	Configuration of a IPv4 DHCP server that runs on the network endpoint.
ipv6DhcpServerConfiguration	Ipv6DhcpServerConfiguration	0..1	aggr	Configuration of a IPv6 DHCP server that runs on the network endpoint.

Table 6.118: DhcpServerConfiguration

Class	**Ipv4DhcpServerConfiguration**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::Ethernet Topology			
Note	Defines the configuration of a IPv4 DHCP server that runs on the network endpoint.			
Base	ARObject,Describable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
addressRangeLowerBound	Ip4AddressString	0..1	attr	Lower range of IP addresses to be issued to DHCP clients. IPv4 Address. Notation: 255.255.255.255.
addressRangeUpperBound	Ip4AddressString	0..1	attr	Upper range of IP addresses to be issued to DHCP clients. Pv4 Address. Notation: 255.255.255.255.
defaultGateway	Ip4AddressString	0..1	attr	IP address of the default gateway. Notation 255.255.255.255
defaultLeaseTime	TimeValue	0..1	attr	Amount of time in seconds that a client may keep the IP address.
dnsServerAddress	Ip4AddressString	*	attr	IP addresses of preconfigured DNS servers. Notation 255.255.255.255 **Tags:** xml.namePlural=DNS-SERVER-ADDRESSES
networkMask	Ip4AddressString	0..1	attr	Default network mask to be used by DHCP clients. Notation 255.255.255.255

Table 6.119: Ipv4DhcpServerConfiguration

Class	**Ipv6DhcpServerConfiguration**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::Ethernet Topology			
Note	Defines the configuration of a IPv6 DHCP server that runs on the network endpoint.			
Base	ARObject,Describable			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
addressRangeLowerBound	Ip6AddressString	0..1	attr	Lower range of IP addresses to be issued to DHCP clients. IPv6 Address. Notation: FFFF:...:FFFF.
addressRangeUpperBound	Ip6AddressString	0..1	attr	Upper range of IP addresses to be issued to DHCP clients. IPv6 Address. Notation: FFFF:...:FFFF.
defaultGateway	Ip6AddressString	0..1	attr	IP address of the default gateway. Notation 255.255.255.255
defaultLeaseTime	TimeValue	0..1	attr	Amount of time in seconds that a client may keep the IP address.
dnsServerAddress	Ip6AddressString	*	attr	IP addresses of preconfigured DNS servers. Notation: FFFF:...:FFFF. **Tags:** xml.namePlural=DNS-SERVER-ADDRESSES
networkMask	Ip6AddressString	0..1	attr	Default network mask to be used by DHCP clients. Notation 255.255.255.255

Table 6.120: Ipv6DhcpServerConfiguration

The `TimeSyncServerConfiguration` provides a time synchronization service. The consuming `TimeSyncClientConfiguration` has to specify the same `timeSyncTechnology` as the server.

Class	**TimeSynchronization**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::Ethernet Topology			
Note	Defines the servers / clients in a time synchronised network.			
Base	ARObject			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
timeSyncClient	TimeSyncClientConfiguration	0..1	aggr	Configuration of the time synchronisation client.
timeSyncServer	TimeSyncServerConfiguration	0..1	aggr	Configuration of the time synchronisation server.

Table 6.121: TimeSynchronization

Class	**TimeSyncClientConfiguration**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::Ethernet Topology			
Note	Defines the configuration of the time synchronisation client.			
Base	ARObject			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*

Attribute	Datatype	Mul.	Kind	Note
ordered Master (ordered)	OrderedMaster	*	aggr	Defines a list of ordered NetworkEndpoints. **Tags:** xml.namePlural=ORDERED-MASTER-LIST
timeSyncTechnology	TimeSyncTechnologyEnum	1	attr	Defines the time synchronisation technology used.

Table 6.122: TimeSyncClientConfiguration

Class	TimeSyncServerConfiguration			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Defines the configuration of the time synchronisation server.			
Base	ARObject,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
priority	PositiveInteger	0..1	attr	Server Priority.
syncInterval	TimeValue	1	attr	Synchronisation interval used by the time synchronisation server (in seconds).
timeSyncServerIdentifier	String	0..1	attr	Identifier of the TimeSyncServer.
timeSyncTechnology	TimeSyncTechnologyEnum	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 6.123: TimeSyncServerConfiguration

Class	OrderedMaster			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Element in the network endpoint list.			
Base	ARObject			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
index	PositiveInteger	1	attr	Defines the order of the network endpoint list (e.g. 0, 1, 2, ...).
timeSyncServer	TimeSyncServerConfiguration	1	ref	Reference to a master (Time Sync Server).

Table 6.124: OrderedMaster

Enumeration	TimeSyncTechnologyEnum
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology
Note	Timesynchronization. Server/Client configuration.
Literal	***Description***

avb_ieee802_1AS	Ethernet AVB compliant IEEE802.1AS Precision Time Protocol
ntp_rfc958	Network Time Protocol (NTP)
ptp_ieee1588_2002	Precision Time Protocol (PTP) IEEE 1588-2002
ptp_ieee1588_2008	Precision Time Protocol (PTP) IEEE 1588-2008

Table 6.125: TimeSyncTechnologyEnum

The DoIpEntity (Diagnostics over Internet Protocol, ISO 13400) defines the DoIp role this NetworkEndpoint has.

Class	**DoIpEntity**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	ECU providing this infrastructure service is a DoIP-Entity.			
Base	ARObject			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
doIpEntityRole	DoIpEntityRoleEnum	1	attr	Identifies the role in terms of DoIP this network-node has.

Table 6.126: DoIpEntity

Enumeration	**DoIpEntityRoleEnum**
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology
Note	DoIP role a network-node has.
Literal	***Description***
edgeNode	Network node is a DoIP gateway that accepts external connections.
gateway	Network node is a Gateway between the DoIP network and other networks.
node	Network node is a DoIp node.

Table 6.127: DoIpEntityRoleEnum

6.7.5.3 Application Endpoint

An ApplicationEndpoint is the endpoint on an EcuInstance in terms of application addressing. The NetworkEndpoint that is related to the application address shall be derived from the aggregating SocketAddress. The SocketAddress connects the IP-address with the transport layer.

[TPS_SYST_01089] ApplicationEndpoint priority ⌈ The priority at the ApplicationEndpoint shall be used as Ethernet Header information together with the vlanIdentifier. If defined the priority overwrites the defaultPriority that is defined in the VlanMembership and the priority that is defined at the NetworkEndpoint. The priority zero represents the highest priority. ⌋(RS_SYST_00039)

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	***Datatype***	***Mul.***	***Kind***	***Note***
consumedServiceInstance	ConsumedServiceInstance	*	aggr	Consumed service instances.
discoveryTechnology	DiscoveryTechnology	0..1	aggr	Defines the used Service-Discovery protocol.
networkEndpoint	NetworkEndpoint	1	ref	Reference to the network address.
priority	PositiveInteger	0..1	attr	Priority defined per application endpoint
providedServiceInstance	ProvidedServiceInstance	*	aggr	Provided service instances.
remotingTechnology	RemotingTechnology	0..1	aggr	Defines the used remoting Technology.
serializationTechnology	SerializationTechnology	0..1	ref	Defines the used serialization technology.
tpConfiguration	TransportProtocolConfiguration	0..1	aggr	Configuration of the used transport protocol.

Table 6.128: ApplicationEndpoint

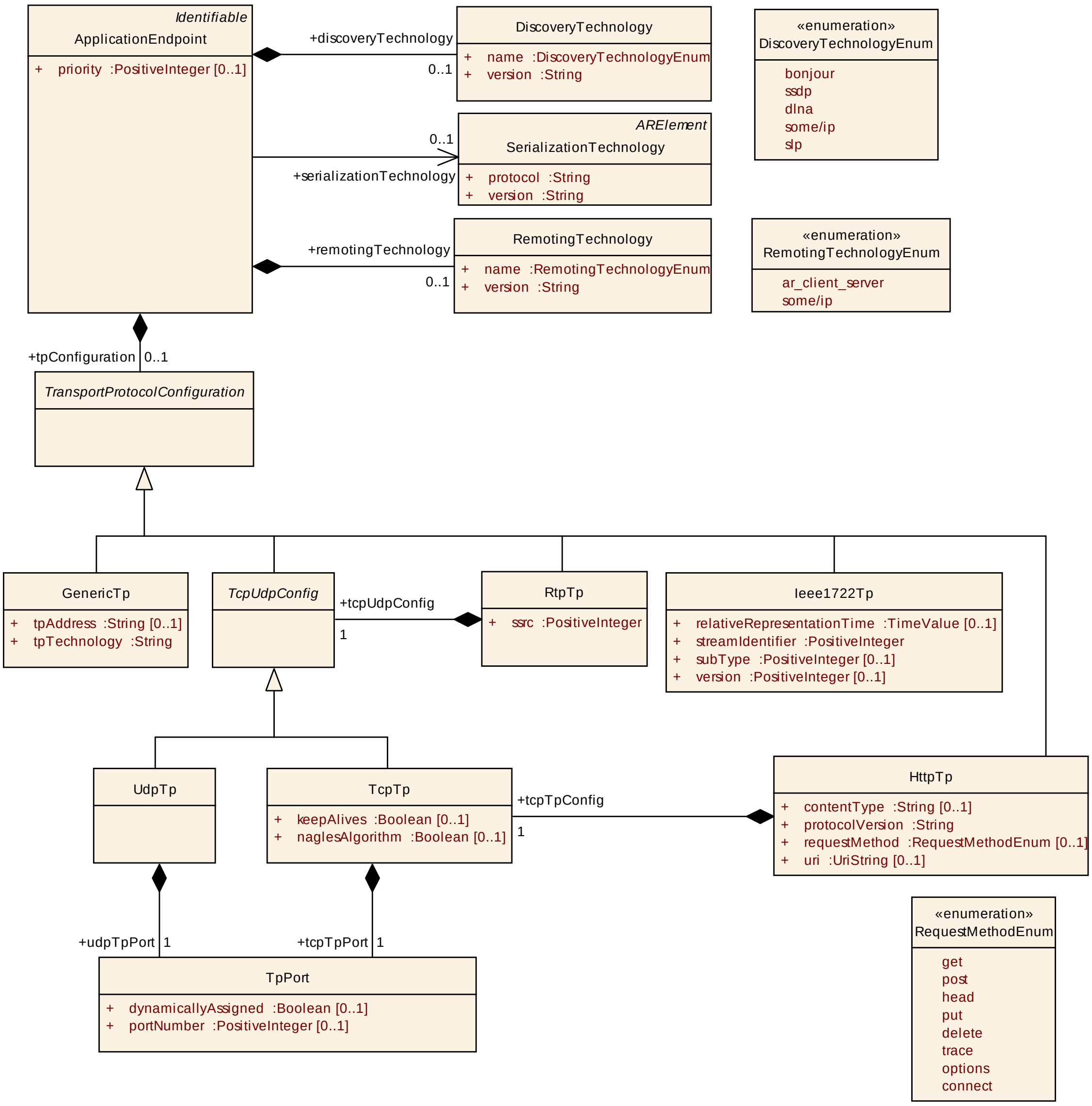

Figure 6.29: Application Endpoint

Class	**TransportProtocolConfiguration (abstract)**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::Ethernet Topology			
Note	Transport Protocol configuration.			
Base	ARObject			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 6.129: TransportProtocolConfiguration

The following Transport Protocols are supported by the System Template:

Class	GenericTp			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Content Model for a generic transport protocol.			
Base	ARObject,TransportProtocolConfiguration			
Attribute	Datatype	Mul.	Kind	Note
tpAddress	String	0..1	attr	Transport Protocol dependent Address.
tpTechnology	String	1	attr	Name of the used Transport Protocol.

Table 6.130: GenericTp

Class	TcpUdpConfig (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Tcp or Udp Transport Protocol Configuration.			
Base	ARObject,TransportProtocolConfiguration			
Attribute	Datatype	Mul.	Kind	Note
–	–	–	–	–

Table 6.131: TcpUdpConfig

Class	UdpTp			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Content Model for UDP configuration.			
Base	ARObject,TcpUdpConfig,TransportProtocolConfiguration			
Attribute	Datatype	Mul.	Kind	Note
udpTpPort	TpPort	1	aggr	Udp Port configuration.

Table 6.132: UdpTp

Class	TcpTp			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Content Model for TCP configuration.			
Base	ARObject,TcpUdpConfig,TransportProtocolConfiguration			
Attribute	Datatype	Mul.	Kind	Note
keepAlives	Boolean	0..1	attr	Indicates if Keep-Alive messages are send.
naglesAlgorithm	Boolean	0..1	attr	Indicates if Nagle's Algorithm is used.
tcpTpPort	TpPort	1	aggr	TCP Port configuration.

Table 6.133: TcpTp

Class	**RtpTp**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::Ethernet Topology			
Note	RTP over UDP or over TCP as transport protocol.			
Base	ARObject,TransportProtocolConfiguration			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
ssrc	PositiveInteger	1	attr	Synchronization source identifier uniquely identifies the source of a stream. The synchronization sources within the same RTP session will be unique.
tcpUdpConfig	TcpUdpConfig	1	aggr	Tcp or Udp Configuration.

Table 6.134: RtpTp

Class	**Ieee1722Tp**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::Ethernet Topology			
Note	Content Model for IEEE 1722 configuration.			
Base	ARObject,TransportProtocolConfiguration			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
relativeRepresentationTime	TimeValue	0..1	attr	Defines the time when content shall be presented (in seconds). The actual absolute time is creation time plus relative presentation time.
streamIdentifier	PositiveInteger	1	attr	IEEE 1722 stream identifier
subType	PositiveInteger	0..1	attr	Protocol type.
version	PositiveInteger	0..1	attr	Revision of Ieee1722 standard

Table 6.135: Ieee1722Tp

Class	**HttpTp**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::Ethernet Topology			
Note	Http over TCP as transport protocol.			
Base	ARObject,TransportProtocolConfiguration			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
contentType	String	0..1	attr	Descriptor for the transported content.
protocolVersion	String	1	attr	HTTP Protocol version (e.g. 1.1)
requestMethod	RequestMethodEnum	0..1	attr	HTTP request method to be used.
tcpTpConfig	TcpTp	1	aggr	TcpTp Configuration.
uri	UriString	0..1	attr	URI to be called.

Table 6.136: HttpTp

Class	TpPort			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Dynamic or direct assignment of a PortNumber.			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
dynamicallyAssigned	Boolean	0..1	attr	Indicates whether the source port is dynamically assigned.
portNumber	PositiveInteger	0..1	attr	Port Number.

Table 6.137: TpPort

[constr_3063] Usage of portNumber and dynamicallyAssigned with value "true" is mutually exclusive ⌈ Usage of portNumber and dynamicallyAssigned with value "true" is mutually exclusive. ⌋

In addition to the TransportProtocolConfiguration the used remoting Technology, serialization Technology and discovery Technology can be described.

- The session layer (Layer 5) is related to the applied Inter Process Communication protocol. This is for the logical control of the communication. The RemotingTechnology describes the related technologies.
- The presentation layer (Layer 6) describes the internal representation of data. SerializationTechnology describes the payload serialization.
- Service Discovery protocols (Layer 7) (DiscoveryTechnology) announce the provided and consumed services.

Class	DiscoveryTechnology			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	This element is deprecated and will be removed in future. This information is replaced by the runtimePortConfiguration and runtimeIpAddressConfiguration attributes in the SocketConnection. Old description: Discovery technology information. **Tags:** atp.Status=obsolete; atp.StatusRevisionBegin=4.1.2			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
name	DiscoveryTechnologyEnum	1	attr	Discovery technology used.
version	String	1	attr	Version of the used Discovery protocol.

Table 6.138: DiscoveryTechnology

Enumeration	**DiscoveryTechnologyEnum**
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology
Note	Discovery technology information.
Literal	*Description*
bonjour	Bonjour Service Discovery
dlna	DLNA UPnP Device Control Protocol Framework
slp	Service Location Protocol
someip	Header format to be used with Remote Procedure Call (RPC) Messages in Client/Server-Communication as well as Sender/Receiver Messages.
ssdp	Simple Service Discovery Protocol (SSDP)

Table 6.139: DiscoveryTechnologyEnum

Class	**RemotingTechnology**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Defines the used remoting Technology.			
Base	ARObject			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
name	RemotingTechnologyEnum	1	attr	Remoting technology used.
version	String	1	attr	Version of the used remoting Technology.

Table 6.140: RemotingTechnology

Enumeration	**RemotingTechnologyEnum**
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology
Note	Remoting technology identifier.
Literal	*Description*
ar_client_server	
someip	

Table 6.141: RemotingTechnologyEnum

In addition the ApplicationEndpoint may operate as a provider (ProvidedServiceInstance) or a consumer (ConsumedServiceInstance) of a service instance. A service represents a functional entity that offers an interface. This interface can be provided by multiple Software Components within an AUTOSAR ECU. To create the connection to the VFB View the service instances in the System Template may be referenced by one or several different DataMappings.

A ProvidedServiceInstance may receive requests from the ConsumedServiceInstance and respond to them. This is realized by ClientServerOperations.

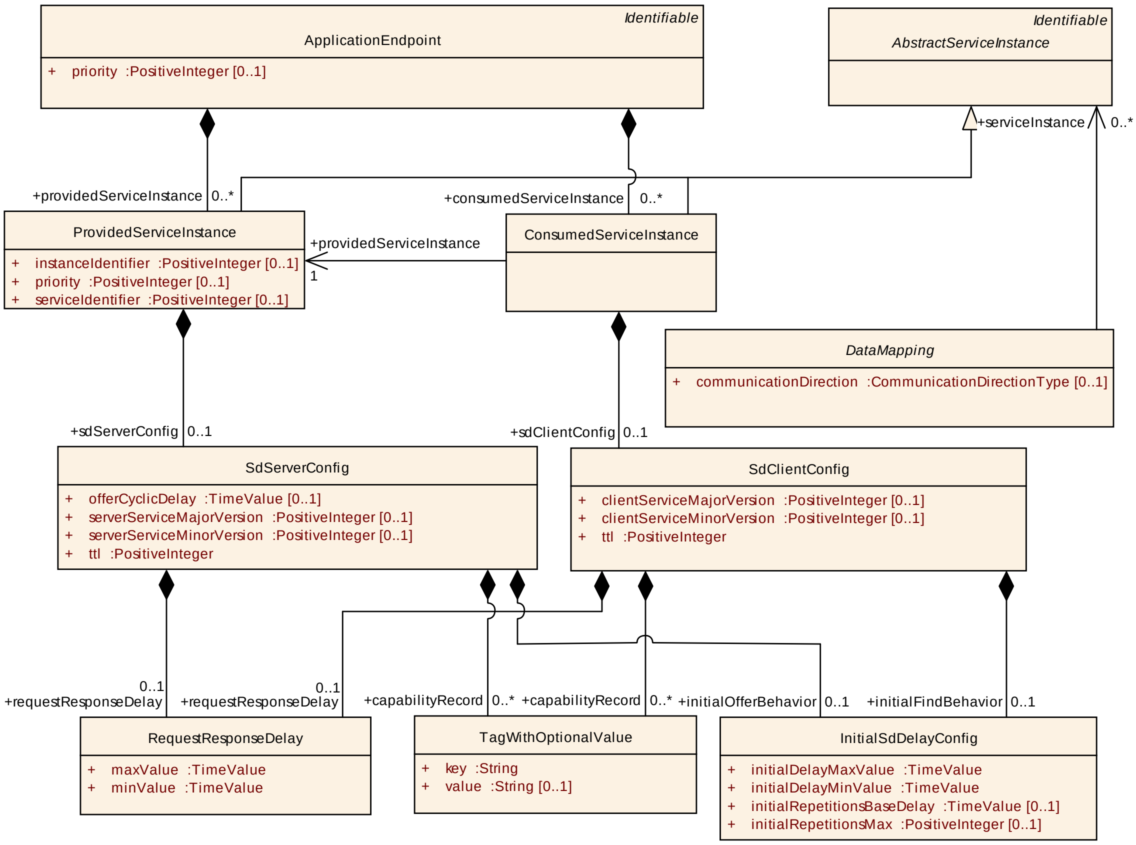

Figure 6.30: Service Instances

Class	**AbstractServiceInstance (abstract)**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Provided and Consumed Ethernet Service Instances that are available at the ApplicationEndpoint.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
routingGroup	SoAdRoutingGroup	*	ref	The ServiceDiscovery module is able to activate and deactivate the PDU routing from and to TCP/IP-sockets.

Table 6.142: AbstractServiceInstance

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	**Datatype**	**Mul.**	**Kind**	**Note**
EventHandler	EventHandler	*	aggr	Collection of event callback configurations.

Attribute	Datatype	Mul.	Kind	Note
instanceIdentifier	PositiveInteger	0..1	attr	Instance identifier. Can be used for e.g. service discovery to identify the instance of the service.
priority	PositiveInteger	0..1	attr	
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 6.143: ProvidedServiceInstance

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	**Datatype**	**Mul.**	**Kind**	**Note**
consumedEventGroup	ConsumedEventGroup	*	aggr	Selection of event-groups the consumer wants to subscribe for.
providedServiceInstance	ProvidedServiceInstance	1	ref	Reference to a providedServiceInstance to get the instanceIdentifier information from the ProvidedServiceInstance.
sdClientConfig	SdClientConfig	0..1	aggr	Service Discovery Client configuration.

Table 6.144: ConsumedServiceInstance

[TPS_SYST_01108] `ProvidedServiceInstance` priority ⌈ The `priority` in the `ProvidedServiceInstance` shall be used as Ethernet Header information together with the `vlanIdentifier`. If defined the `priority` overwrites the `defaultPriority` that is defined in the `VlanMembership`, the `priority` that is defined at the `NetworkEndpoint` and the `priority` that is defined at the `ApplicationEndpoint`. The priority zero represents the highest priority. ⌋*(RS_SYST_00039)*

The AUTOSAR BswM is used to aggregate the availability of all entities which make up a service instance. When all entities are available, the service instance as such is available. When a service instance becomes available the SD Module will usually send an announcement message in order for other ECUs to learn about the availability and the location (IP address and UDP or TCP Port) of that service instance.

The Service Discovery configuration in the System Template is described by the two elements `SdServerConfig` and `SdClientConfig`:

Class	**SdServerConfig**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Server configuration for Service-Discovery.			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
capabilityRecord	TagWithOptionalValue	*	aggr	A sequence of records to store arbitrary name/value pairs conveying additional information about the named service.
initialOfferBehavior	InitialSdDelayConfig	0..1	aggr	Controls offer behavior of the server.
offerCyclicDelay	TimeValue	0..1	attr	Optional attribute to define cyclic offers. Cyclic offer is active, if the delay is set (in seconds).
requestResponseDelay	RequestResponseDelay	0..1	aggr	Maximum/Minimum allowable response delay to entries received by multicast in seconds.
serverServiceMajorVersion	PositiveInteger	0..1	attr	Major version number of the Service.
serverServiceMinorVersion	PositiveInteger	0..1	attr	Minor version number of the Service.
ttl	PositiveInteger	1	attr	Time to live. Shall be a positive value (sInt32).

Table 6.145: SdServerConfig

Class	**SdClientConfig**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Client configuration for Service-Discovery.			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
capabilityRecord	TagWithOptionalValue	*	aggr	A sequence of records to store arbitrary name/value pairs conveying additional information about the named service.
clientServiceMajorVersion	PositiveInteger	0..1	attr	Major version number of the Service.
clientServiceMinorVersion	PositiveInteger	0..1	attr	Minor version number of the Service.
initialFindBehavior	InitialSdDelayConfig	0..1	aggr	Controls initial find behavior of clients.
requestResponseDelay	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 6.146: SdClientConfig

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.			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
key	String	1	attr	Defines a key.
value	String	0..1	attr	Defines the corresponding value.

Table 6.147: TagWithOptionalValue

[TPS_SYST_01094] allowed `key`/`value` `TagWithOptionalValue` combinations
⌈ The following `key`/`value` combinations are supported:

- `key` present, with no `value` (e.g. "passreq" -> password required for this service)
- `key` present, with empty `value` (e.g. "PlugIns=" -> server supports plugins, but none are presently installed)
- `key` present, with non-empty `value` (e.g. "PlugIns=JPEG,MPEG2,MPEG4")

⌋(*RS_SYST_00039*)

Class	RequestResponseDelay			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	Time to wait before answering the query.			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
maxValue	TimeValue	1	attr	Maximum allowable response delay to entries received by multicast in seconds.
minValue	TimeValue	1	attr	Minimum allowable response delay to entries received by multicast in seconds.

Table 6.148: RequestResponseDelay

Class	InitialSdDelayConfig			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology			
Note	This element is used to configure the offer behavior of the server and the find behavior on the client.			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
initialDelayMaxValue	TimeValue	1	attr	Max Value in seconds to delay randomly the first offer (if aggregated by SdServerConfig) or the transmission of a find message (if aggregated by SdClientConfig).

Attribute	Datatype	Mul.	Kind	Note
initialDelayMinValue	TimeValue	1	attr	Min Value in seconds to delay randomly the first offer (if aggregated by SdServerConfig) or the transmission of a find message (if aggregated by SdClientConfig).
initialRepetitionsBaseDelay	TimeValue	0..1	attr	The base delay for offer repetitions (if aggregated by SdServerConfig) or find repetitions (if aggregated by SdClientConfig. Successive find messages have an exponential back off delay.
initialRepetitionsMax	PositiveInteger	0..1	attr	Describes the maximum amount of offer repetitions (if aggregated by SdServerConfig) or the maximum amount of find repetitions (if aggregated by SdClientConfig).

Table 6.149: InitialSdDelayConfig

There are use cases where the client wants to be notified about events that occur at the server without the need to make an explicit request. A ConsumedServiceInstance can subscribe to event groups that are modeled as ConsumedEventGroups. All event/notification consumers (ConsumedEventGroups) are referenced by the EventHandler. The notification is described in the VFB view with VariableDataPrototypes that are sent via a sender/receiver interface from the ProvidedServiceInstance to all event/notification consumers. At the ConsumedServiceInstance the event-callback is processed and has normally a void return.

The availability of a consumer to receive events is configured with the SdClientConfig. The configuration of the EventHandler with the Service Discovery client attributes (SdServerConfig) ensures that the EventHandler knows which consumers are available and to which consumer the notification can be send.

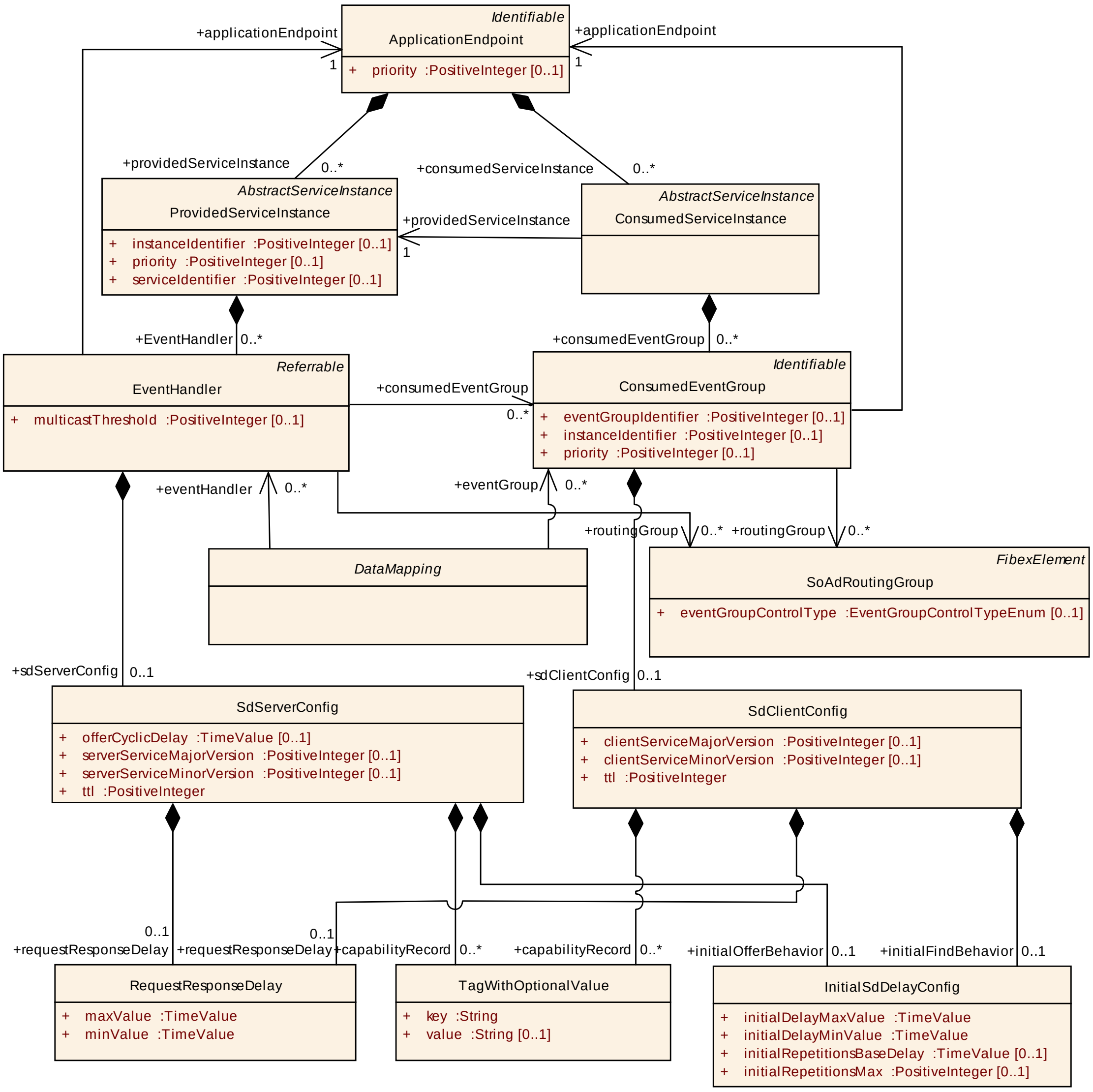

Figure 6.31: Event Handler

Class	**EventHandler**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::Ethernet Topology			
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	***Datatype***	***Mul.***	***Kind***	***Note***
application Endpoint	ApplicationEndpoint	1	ref	Defines the local application endpoint used to submit an event to a subscriber. For the submission of events the service provider may use a different TpPort address (ApplicationEndpoint) then for the response of requests.
consumed EventGroup	ConsumedEventGroup	*	ref	All consumers of the event are referenced here.

Attribute	Datatype	Mul.	Kind	Note
multicastThreshold	PositiveInteger	0..1	attr	Specifies the number of subscribed clients that trigger the server to change the transmission of events to multicast. If configured to 0 only unicast will be used. If configured to 1 the first client will be already served by multicast. If configured to 2 the first client will be server with unicast and as soon as the second client arrives both will be served by multicast. This does not influence the handling of initial events, which are served using unicast only.
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 6.150: EventHandler

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	**Datatype**	**Mul.**	**Kind**	**Note**
applicationEndpoint	ApplicationEndpoint	1	ref	Defines the application endpoint where the events of the event group are received. This may be a different TpPort address (ApplicationEndpoint) then that which is used for the sending of requests.
eventGroupIdentifier	PositiveInteger	0..1	attr	EventGroup ID. Shall be unique within one system to allow service discovery.
instanceIdentifier	PositiveInteger	0..1	attr	Instance identifier. Can be used for e.g. service discovery to identify the instance of the event group.
priority	PositiveInteger	0..1	attr	Priority defined per consumed Event-Group
routingGroup	SoAdRoutingGroup	*	ref	The ServiceDiscovery module is able to activate and deactivate the PDU routing for receiving events.
sdClientConfig	SdClientConfig	0..1	aggr	The readiness to receive events is defined by the ServiceDiscovery of the ConsumedEventGroup. The EventHandler shall know about this announcement to decide about the submission of events. Therefore the EventHandler may be configured with Service-Discovery Client attributes.

Table 6.151: ConsumedEventGroup

[TPS_SYST_01151] DataMapping reference to an EventHandler ⌈ If the DataMapping references an EventHandler in the role eventHandler the serviceInstance reference to the ProvidedServiceInstance or ConsumedServiceInstance that aggregates this specific EventHandler could be skipped. ⌋

[TPS_SYST_01152] DataMapping reference to a ConsumedEventGroup ⌈ If the DataMapping references a ConsumedEventGroup in the role eventGroup the serviceInstance reference to the ProvidedServiceInstance or ConsumedServiceInstance that aggregates this specific ConsumedEventGroup could be skipped. ⌋

6.7.5.4 Diagnostics over IP

In AUTOSAR the DoIp functionality is implemented in the same software module as the SoAd functionality. It will encapsulate, split and reassemble data and manage connections using the DoIP wrapper specification and thus has the functionality of a transport protocol similar to the CAN-TP or FlexRay-TP. It utilizes the socket interface of the TCP/IP stack module to send and receive UDP data packets and receive and transmit data streams through TCP-sockets. Therefore the model from figure 6.22 shall be used to describe DoIp in the System Description. The optional references to the LogicAddress are only relevant for the DoIp description.

Class	**LogicAddress**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Ethernet::Ethernet Communication			
Note	The logical DoIP address. This element shall only be used for DoIP (Diagnosis over IP).			
Base	ARObject,Identifiable,MultilanguageReferrable,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
address	Integer	1	attr	The logical DoIP address.

Table 6.152: LogicAddress

6.8 Transport Layer

In AUTOSAR, the Transport Layer has two main purposes: The segmentation and reassembly of messages that are too long to fit into one frame on the underlying communication cluster, and the re-use of fixed frame identifiers for different message content.

According to the AUTOSAR Layered Software Architecture [14], each type of communication cluster has its own definition of the Transport Layer. Consequently, the peculiarities of the cluster types are addressed in the System Template by having different detailed models for FlexRay, CAN, LIN and J1939. However, all models are embedded into the communication model: They use specialized classes of TpConfig as a root element into the TP configuration.

[TPS_SYST_01099] Context of TpConfig ⌈ A TpConfig element is existing always in the context of exactly one CommunicationCluster. ⌋*(RS_SYST_00014)*

All Transport Layers will take IPdus as input elements, which will be transferred in the form of one or more NPdus. A TpConnection (FlexrayTpConnection, CanTpConnection, LinTpConnection, J1939TpConnection) identifies a connection link between different communication nodes and routes the Pdus between them.

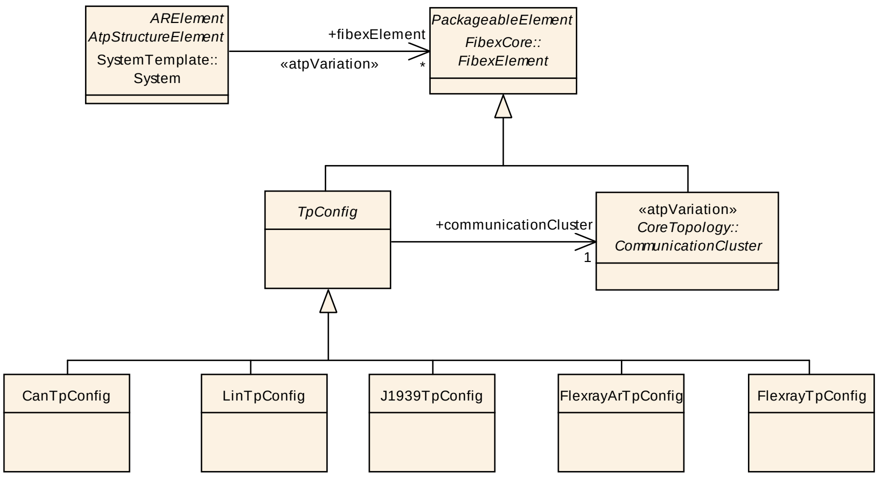

Figure 6.32: Transport Layer Overview

Examples in chapter 6.8.7 and chapter 6.8.8 illustrate the usage of the TP model.

Class	**TpConfig (abstract)**			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	Contains all configuration elements for AUTOSAR TP.			
Base	ARObject,CollectableElement,FibexElement,Identifiable,Multilanguage Referrable,PackageableElement,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
communicationCluster	CommunicationCluster	1	ref	A TpConfig is existing always in the context of exactly one CommunicationCluster.

Table 6.153: TpConfig

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,PackageableElement,Pdu,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 6.154: NPdu

Class	TpAddress			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	An ECUs TP address on the referenced channel. This represents the diagnostic Address.			
Base	ARObject,Identifiable,MultilanguageReferrable,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
tpAddress	Integer	1	attr	An ECUs TP address on the referenced channel. This represents the diagnostic Address.

Table 6.155: TpAddress

[constr_3025] Usage of `NPdu`s in `TpConnections` ⌈ In case several `TpConnections` 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). ⌋

Note: Depending on the capabilities of the Basic Software implementations of Tp and Interface the ECU Configuration of the respective BSW Modules may utilize more communication elements (`NPdu`s).

Example for an allowed System Template description where the same FrameId is used by two different `TpConnection`s:

```
TpConnection1 --(dataPdu)--> NPdu1 ----> FrameId1
TpConnection1 --(flowControl)--> NPdu2 ----> FrameId2
TpConnection2 --(dataPdu)--> NPdu2 ----> FrameId2
TpConnection2 --(flowControl)--> NPdu1 ----> FrameId1
```

The following Ecu configuration with additional `NPdu`s can still be derived from the above system description:

```
TpConnection1 --(dataPdu)--> NPdu1 ----> FrameId1
TpConnection1 --(flowControl)--> NPdu2 ----> FrameId2
TpConnection2 --(dataPdu)--> NPdu3 ----> FrameId2
TpConnection2 --(flowControl)--> NPdu4 ----> FrameId1
```

[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. ⌋

The corresponding PduTriggering for the IPdu referenced from the TpConnection in the role tpSdu is aggregated by the PhysicalChannel which points to the same CommunicationConnector which is referenced by TpNode that this TpConnection points to.

Please note that with [constr_3090] the multiple transmission of the same TpSdu over a specific channel using TP is only possible if several IPdus and TpConnections are created.

6.8.1 Transport Layer Routing

Pdu routing is only supported for IPdus. The transformations in the TP modules take a significant amount of time and resources.

[TPS_SYST_01100] TP routing using the same transport protocol (low-level routing) ⌈ The behavior can be optimized if source and target use the same transport protocol (e.g. CanTp-to-CanTp routing). In this case the inbound NPdu can be directly forwarded to the PduR and then sent on the outbound bus without any (resource consuming) TP module involvement. ⌋*(RS_SYST_00014)*

To support such a "low-level" TP routing in the System Template the NPdu element is a specialization of the IPdu element. This allows the PDU-routing of NPdus. [4]

[TPS_SYST_01101] TP routing using different transport protocols ⌈ In case of a gateway between different transport protocols every incoming NPdu needs to be:

- forwarded to corresponding inbound TP module and transformed into an IPdu
- the IPdu needs to be forwarded to the PduR
- the PduR routes the IPdu to the outgoing TP module
- the outbound TP module transforms the IPdu into a NPdu which is then sent on the target bus.

⌋*(RS_SYST_00014)*

6.8.2 FlexRay ISO Transport Layer

The FlexRay ISO 10681-2 Transport Layer supports multiple sessions, i.e. multiple segmented transfers can be handled at the same time. Thus, multiple FlexrayTpConnections can be defined on the same ECU. Each FlexrayTpConnection is controlled by configuration parameters defined in FlexrayTpConnectionControl.

[4] A "low-level" PDU-routing is also possible for XcpPdus.

[TPS_SYST_01102] **FlexrayTpConnectionControl** **reuse** ⌈ The same FlexrayTpConnectionControl may be reused for an arbitrary number of FlexrayTpConnections. ⌋*(RS_SYST_00014)*

A FlexrayTpConnection defines the way of communication between a sender and a receiver and uses a FlexrayTpPduPool of NPdus to transmit data to the FlexRay Interface.

[TPS_SYST_01103] **FlexrayTpConnection shall specify one txPduPool** ⌈ Each FlexrayTpConnection shall specify one txPduPool with at least one nPdu. ⌋*(RS_SYST_00014)*

In order to achieve a higher bandwidth a txPduPool may contain more than one transmit NPdu, e.g. if all referenced NPdus are transmitted in different FlexrayFrames in the same cycle.

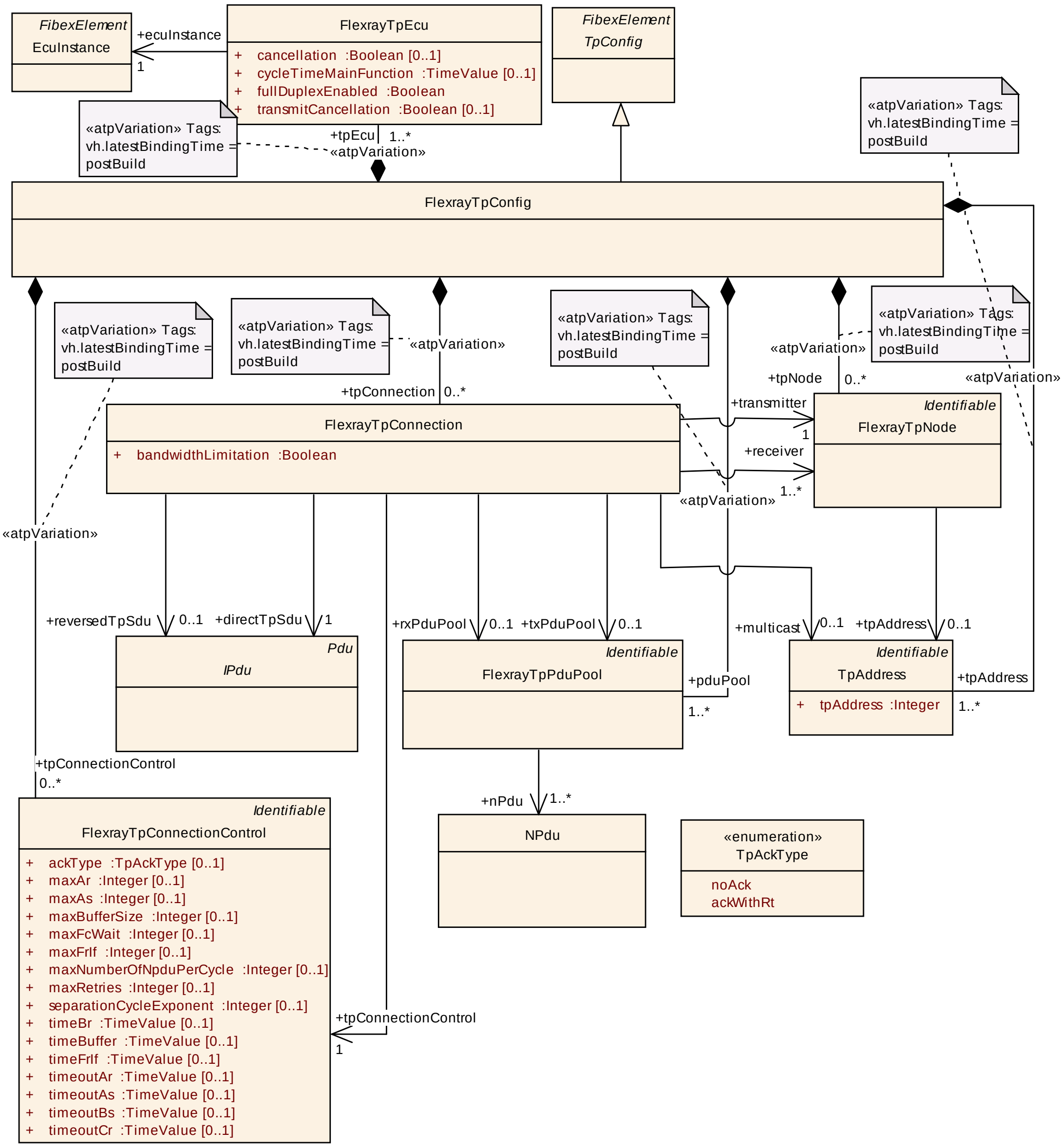

Figure 6.33: FlexRay ISO Transport Layer Configuration (TransportProtocols: FlexRay-IsoTransportProtocol)

FlexrayTpConnections are specifically used for communication between one source and one or several target devices. These communication partners are specified using the transmitter and receiver associations to FlexrayTpNodes, providing the diagnostic tpAddress and the connection to the topology.

[TPS_SYST_01104] FlexrayTpConnection with several receivers ⌈ In case of several receivers a multicast tpAddress shall be used. ⌋*(RS_SYST_00014)*

The actual payload to be transported by the FlexrayTpConnection is specified by using either one or two references to IPdus, depending on whether the connection shall be used unidirectional (one reference) or bidirectional (two references).

Class	**FlexrayTpConfig**			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	This element defines exactly one FlexRay ISO TP Configuration. One FlexRayTpConfig element shall be created for each FlexRay Network in the System that uses FlexRay Iso Tp. **Tags:** atp.recommendedPackage=TpConfigs			
Base	ARObject,CollectableElement,FibexElement,Identifiable,Multilanguage Referrable,PackageableElement,Referrable,TpConfig			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
pduPool	FlexrayTpPduPool	1..*	aggr	Configuration of FlexRay TP Pdu Pools. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
tpAddress	TpAddress	1..*	aggr	Collection of TpAddresses. atpVariation: Derived, because EcuInstance can vary. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
tpConnection	FlexrayTpConnection	*	aggr	Configuration of FlexRay TP Connections. atpVariation: Derived, because TpNode can vary. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
tpConnectionControl	FlexrayTpConnectionControl	*	aggr	Configuration of FlexRay TP Connection Controls. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild

Attribute	Datatype	Mul.	Kind	Note
tpEcu	FlexrayTpEcu	1..*	aggr	Collection of TP Ecus atpVariation: Derived, because EcuInstance can vary. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
tpNode	FlexrayTpNode	*	aggr	Senders and receivers of FlexRay TP messages. atpVariation: Derived, because EcuInstance can vary. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild

Table 6.156: FlexrayTpConfig

Class	**FlexrayTpConnectionControl**			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	Configuration parameters to control a FlexRay TP connection.			
Base	ARObject,Identifiable,MultilanguageReferrable,Referrable			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
ackType	TpAckType	0..1	attr	This parameter defines the type of acknowledgement which is used for the specific channel.
maxAr	Integer	0..1	attr	This parameter defines the maximum number of trying to send a frame when a TIMEOUT AR occurs (depending on whether retry is configured).
maxAs	Integer	0..1	attr	This parameter defines the maximum number of trying to send a frame when a TIMEOUT AS occurs (depending on whether retry is configured)
maxBuffer Size	Integer	0..1	attr	This parameter is only relevant when having retry activated. It limits the maximal buffer size the FrTp can choose in order to limit the amount of Tx buffer that will be requested at the sender side in a segmented transfer.
maxFcWait	Integer	0..1	attr	This attribute defines the maximum number of FlowControl N-PDUs with FlowState "WAIT".
maxFrIf	Integer	0..1	attr	This parameter defines the maximum number of trying to send a frame when the FrIf returns an error
maxNumb erOfNpduP erCycle	Integer	0..1	attr	This parameter limits the number of N-Pdus the sender is allowed to transmit within a FlexRay cycle.
maxRetrie s	Integer	0..1	attr	This parameter defines the maximum number of retries (if retry is configured for the particular channel).
separation CycleExpo nent	Integer	0..1	attr	Exponent to calculate the minimum number of "Separation Cycles" the sender has to wait for the next transmission of an FrTp N-Pdu.

Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
timeBr	TimeValue	0..1	attr	Time (in seconds) until transmission of the next FlowControl N-PDU.
timeBuffer	TimeValue	0..1	attr	This parameter defines the time of waiting for the next try to get a Tx or Rx buffer. This parameter is equivalent to the temporal distance between two FC.WT N-Pdus in case the buffer request returns busy. Specified in seconds.
timeFrIf	TimeValue	0..1	attr	This parameter defines the time of waiting for the next try to send. Specified in seconds.
timeoutAr	TimeValue	0..1	attr	This parameter states the timeout between the PDU transmit request of the Transport Layer to the FlexRay Interface and the corresponding confirmation of the FlexRay Interface on the receiver side (for FC or AF). Specified in seconds.
timeoutAs	TimeValue	0..1	attr	This attribute states the timeout between the PDU transmit request for the first PDU of the group used in the current connection of the Transport Layer to the FlexRay Interface and the corresponding confirmation of the FlexRay Interface (when having sent the last PDU of the group used in this connection) on the sender side (SF-x, FF-x, CF or FC (in case of Transmit Cancellation)). Specified in seconds.
timeoutBs	TimeValue	0..1	attr	This parameter defines the timeout in seconds for waiting for an FC or AF on the sender side in a 1:1 connection.
timeoutCr	TimeValue	0..1	attr	This parameter defines the timeout value in seconds for waiting for a CF or FF-x (in case of retry) after receiving the last CF or after sending an FC or AF on the receiver side. Specified in seconds.

Table 6.157: FlexrayTpConnectionControl

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			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
bandwidth Limitation	Boolean	1	attr	Specifies whether the connection requires a bandwidth limitation or not.

Attribute	Datatype	Mul.	Kind	Note
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 (FrTpRxPduPool) which are defined for receiving data via this particular connection. The following constraint is valid only for the System Extract/ECU Extract: In case this connection is applied to the transmitter the rxPduPool holds the actually received NPdus. In case this connection is applied to the receiver the rxPduPool holds the actually sent NPdus.
tpConnectionControl	FlexrayTpConnectionControl	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 (FrTpTxPduPool) which are defined for sending data via this particular connection. The following constraint is valid only for the System Extract/ECU Extract: In case this connection is applied to the transmitter the txPduPool holds the actually sent NPdus. In case this connection is applied to the receiver the txPduPool holds the actually received NPdus.

Table 6.158: FlexrayTpConnection

The `FlexrayTpConnection` refers to the `FlexrayTpPduPool` in two roles: `rxPduPool` and `txPduPool`.

[TPS_SYST_01064] Transmit/Receive Semantics of Pdu Pools ⌈ The transmit/receive semantics of Pdu Pools depends on the role of the regarded ECU:

- If the ECU is the transmitter then the `txPduPool` holds the sent `NPdu`s and the `rxPduPool` holds the received `NPdu`s.
- If the ECU is the receiver then the the `txPduPool` holds the received `NPdu`s and the `rxPduPool` holds the sent `NPdu`s.

⌋

The following example shows how this differentiation may be used:

System Description:
SENDER = A

RECEIVER = B
TxPool = PDU_1
RxPool = PDU_2

ECU Extract of A:
SENDER = A
TxPool = PDU_1 -> sent Pdus
RxPool = PDU_2 -> received Pdus

Since on receiver side the PDU_1 is received and PDU_2 is sent (from a local point of view) the export shall look like this:
ECU Extract of B:
RECEIVER = B
TxPool = PDU_1 -> received Pdus
RxPool = PDU_2 -> sent Pdus

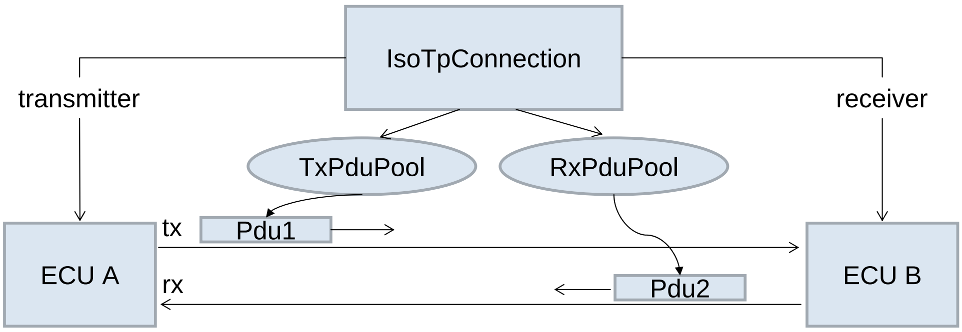

Figure 6.34: IsoTp Example

Class	**FlexrayTpPduPool**			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	FlexrayTpPduPool is a set of N-PDUs which are defined for FrTp sending or receiving purpose.			
Base	ARObject,Identifiable,MultilanguageReferrable,Referrable			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
nPdu	NPdu	1..*	ref	Reference to NPdus that are part of the PduPool.

Table 6.159: FlexrayTpPduPool

Class	**FlexrayTpNode**			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	TP Node (Sender or Receiver) provides the TP Address and the connection to the Topology description.			
Base	ARObject,Identifiable,MultilanguageReferrable,Referrable			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*

Attribute	Datatype	Mul.	Kind	Note
connector	Communication Connector	*	ref	Asscociation to one or more physical connectors (max number of connectors for FlexRay: 2). In a System Description this reference is mandatory. In an ECU Extract this reference is optional (references to ECUs that are not part of the ECU Extract shall be avoided).
tpAddress	TpAddress	0..1	ref	Reference to the TP Address that is used by the TpNode. This reference is optional in case that the multicast TP Address is used (reference from TpConnection).

Table 6.160: FlexrayTpNode

Class	FlexrayTpEcu			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	ECU specific TP configuration parameters. Each TpEcu element has a reference to exactly one ECUInstance in the topology.			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
cancellation	Boolean	0..1	attr	With this switch Tx and Rx Cancellation can be turned on or off.
cycleTimeMainFunction	TimeValue	0..1	attr	The period between successive calls to the Main Function of the AUTOSAR TP. Specified in seconds.
ecuInstance	EcuInstance	1	ref	Connection to the ECUInstance in the Topology
fullDuplexEnabled	Boolean	1	attr	The full duplex mechanisms is enabled if this attribute is set to true. Otherwise half duplex is enabled.
transmitCancellation	Boolean	0..1	attr	This attribute states whether Transmit Cancellation is supported on this ECU. Please note that this attribute is deprecated and will be removed in future. **Tags:** atp.Status=obsolete; atp.StatusComment=replaced by FlexrayTpEcu.cancellation; atp.StatusRevisionBegin=4.1.3

Table 6.161: FlexrayTpEcu

6.8.3 FlexRay AUTOSAR Transport Layer

This section describes a Non-ISO FlexRay TP protocol that is supported by AUTOSAR in addition to the FlexRay ISO 10681-2 TP (see section 6.8.2). The Non-ISO FlexRay Transport Layer supports multiple sessions, i.e. multiple segmented transfers can be handled at the same time.

A FlexrayArTpChannel provides a Tx and an Rx pool of NPdus which are used by the associated FlexrayArTpConnections.

FlexrayArTpConnections are used for communication between one source and one or more target device(s). These communication partners are specified by the source and target associations to FlexrayArTpNodes, providing the diagnostic TpAddresses and the connection to the topology description. The actual payload to be transported by the FlexrayArTpConnection is identified by the references directTpSdu and reversedTpSdu to IPdus. When one of the two SDUs is omitted, the connection shall be used unidirectional.

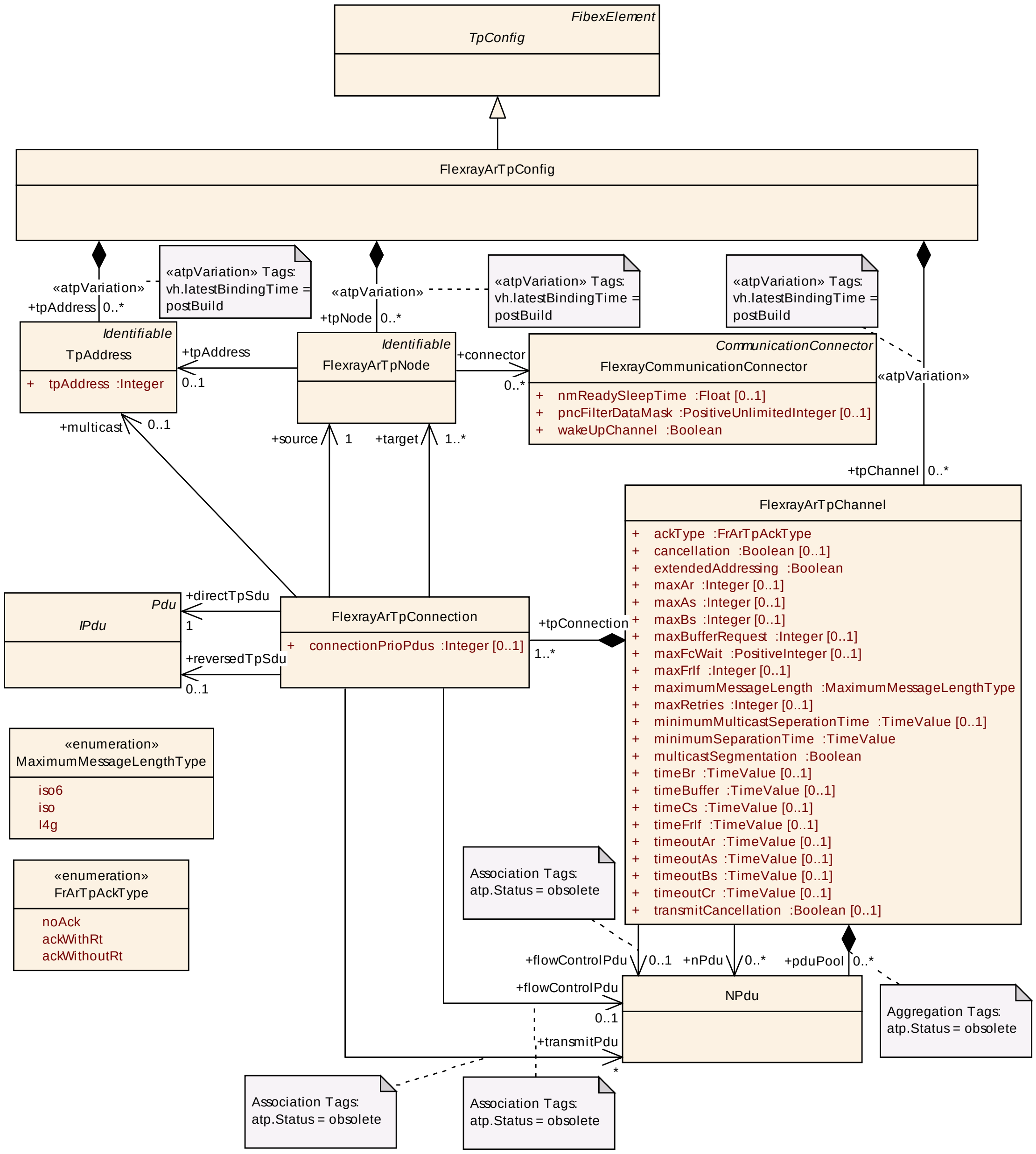

Figure 6.35: FlexRay Autosar Transport Layer Configuration (TransportProtocols: FlexRayAutosarTransportProtocol)

Class	**FlexrayArTpChannel**			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	A channel is a group of connections sharing several properties. The FlexRay AutosarTransport Layer supports several channels. These channels can work concurrently, thus each of them requires its own state machine and management data structures and its own PDU-IDs.			
Base	ARObject			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
ackType	FrArTpAckType	1	attr	Type of Acknowledgement.

Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
cancellation	Boolean	0..1	attr	With this switch Tx and Rx Cancellation can be turned on or off.
extendedAddressing	Boolean	1	attr	Adressing Type of this connection: true: Two Bytes false: One Byte
flowControlPdu	NPdu	0..1	ref	Reference to the Flow Control NPdu. Please note that this reference is deprecated and will be removed in future. **Tags:** atp.Status=obsolete
maxAr	Integer	0..1	attr	This attribute defines the maximum number of trying to send a frame when a TIMEOUT AR occurs (depending on whether retry is configured).
maxAs	Integer	0..1	attr	This attribute defines the maximum number of trying to send a frame when a TIMEOUT AS occurs (depending on whether retry is configured).
maxBs	Integer	0..1	attr	This attribute defines the number of consecutive CFs between two FCs (block size). Valid values are 1 .. 16 when retry is activated, and 0 .. 255 otherwise.
maxBufferRequest	Integer	0..1	attr	Please note that this attribute is deprecated and will be removed in future. maxFcWait will be used instead to configure the maximum number of wait frames on receiver side. On the sender side, timeCs defines the maximum time for retries. **Tags:** atp.Status=obsolete
maxFcWait	PositiveInteger	0..1	attr	This attribute defines the maximal number of wait frames to be sent for a pending connection. Range is 0..255.
maxFrIf	Integer	0..1	attr	Please note that this attribute is deprecated and will be removed in future. Old description: This attribute defines the maximum number of trying to send a frame when the FrIf returns an error. **Tags:** atp.Status=obsolete; atp.StatusRevisionBegin=4.1.2
maxRetries	Integer	0..1	attr	This attribute defines the maximum number of retries (if retry is configured for the particular channel).
maximumMessageLength	MaximumMessageLengthType	1	attr	This specifies the maximum message length for the particular channel.

Attribute	Datatype	Mul.	Kind	Note
minimumMulticastSeparationTime	TimeValue	0..1	attr	This attribute defines the minimum amount of time between two succeeding CFs of a 1:n segmented transmission in seconds. Valid values are 0, 100μs, 200μs ... 900μs, 1ms, 2ms .. 127ms. The value can be changed at runtime using the FrArTp_ChangeParameter interface. minimumMulticastSeparationTime must be an integer multiple of the cycle length multiplied with the multiplexing factor, i.e. minimumMulticastSeparationTime = n * cycle * m, where n is an integer >= 0, cycle is FlexrayCluster.cycle, and m is the cycle multiplexor of those cycles where PDUs of the PDU pool are scheduled. Please note: Due to the scheduling strategies of FrTp, minimumMulticastSeparationTime can only be kept to a degree defined by the maximum temporal distance of the PDUs of a PDU pool within one FlexRay cycle. Range: 0 .. 0.127
minimumSeparationTime	TimeValue	1	attr	This attribute defines the minimum amount of time between two succeeding CFs of a 1:1 segmented transmission in seconds. Valid values are 0, 100μs, 200μs .. 900μs, 1ms, 2ms .. 127ms. The value can be changed at runtime using the FrArTp_ChangeParameter interface. The minimumSeparationTime must be an integer multiple of the cycle length multiplied with the multiplexing factor, i.e. minimumSeparationTime = n * cycle * m, where n is an integer >=0, cycle is FlexrayCluster.cycle, and m is the cycle multiplexor of those cycles where PDUs of the PDU pool are scheduled. Please note: Due to the scheduling strategies of FrTp, minimumSeparationTime can only be kept to a degree defined by the maximum temporal distance of the PDUs of a PDU pool within one FlexRay cycle. Range: 0 .. 0.127
multicastSegmentation	Boolean	1	attr	This attribute defines whether segmentation within a 1:n connection is allowed or not.
nPdu	NPdu	*	ref	A FlexRayTpChannel references a set of NPdus. These NPdus are logically assembled into a pool of Rx NPdus and another pool of Tx NPdus. It must be ensured that a second channel either references all NPdus of such a pool, or none.

Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
pduPool	NPdu	*	aggr	Please note that this aggregation is deprecated and will be removed in future. The nPdu reference will be used instead. **Tags:** atp.Status=obsolete
timeBr	TimeValue	0..1	attr	This attribute defines the time in seconds between receiving the last CF of a block or an FF-x (or SF-x) and sending out an FC or AF.
timeBuffer	TimeValue	0..1	attr	Please note that this attribute is deprecated and will be removed in future. timeBr will be used instead to configure the delay between two wait frames (and thus two buffer requests) on receiver side. On sender side, the main task cycle will be used. **Tags:** atp.Status=obsolete
timeCs	TimeValue	0..1	attr	This attribute defines the time in seconds between the sending of two consecutive frames or between a consecutive frame and a flow control (for Transmit Cancellation) or between reception of an flow control or Acknowledgement Frame and sending of the next consecutive frame or a flow control (for Transmit Cancellation).
timeFrIf	TimeValue	0..1	attr	Please note that this attribute is deprecated and will be removed in future. Old description: This attribute defines the time in seconds of waiting for the next try (if retry is activated) to send via FrIf_Transmit. Specified in seconds. **Tags:** atp.Status=obsolete; atp.StatusRevision Begin=4.1.2
timeoutAr	TimeValue	0..1	attr	This attribute states the timeout in seconds between the PDU transmit request of the Transport Layer to the FlexRay Interface and the corresponding confirmation of the FlexRay Interface on the receiver side (for FC or AF).
timeoutAs	TimeValue	0..1	attr	This attribute states the timeout in seconds between the PDU transmit request for the first PDU of the group used in the current connection of the Transport Layer to the FlexRay Interface and the corresponding confirmation of the FlexRay Interface (when having sent the last PDU of the group used in this connection) on the sender side (SF-x, FF-x, CF).
timeoutBs	TimeValue	0..1	attr	This attribute defines the timeout in seconds for waiting for an FC or AF on the sender side in a 1:1 connection.

Attribute	Datatype	Mul.	Kind	Note
timeoutCr	TimeValue	0..1	attr	This attribute defines the timeout value in seconds for waiting for a CF or FF-x (in case of retry) after receiving the last CF or after sending an FC or AF on the receiver side.
tpConnection	FlexrayArTpConnection	1..*	aggr	Group of connections that can be used in this channel.
transmitCancellation	Boolean	0..1	attr	This attribute states whether Transmit Cancellation is supported on this channel. When not set, the value of this attribute may be specified by the ECU integrator. Please note that this attribute is deprecated and will be removed in future. **Tags:** atp.Status=obsolete; atp.StatusComment=replaced by FrArTpChannel.cancellation; atp.StatusRevisionBegin=4.1.3

Table 6.162: FlexrayArTpChannel

Class	**FlexrayArTpNode**			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	TP Node (Sender or Receiver) provides the TP Address and the connection to the Topology description.			
Base	ARObject,Identifiable,MultilanguageReferrable,Referrable			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
connector	FlexrayCommunicationConnector	*	ref	Association to one or more physical connectors (max number of connectors for FlexRay: 2). In a System Description this reference is mandatory. In an ECU Extract this reference is optional (references to ECUs that are not part of the ECU Extract shall be avoided).
tpAddress	TpAddress	0..1	ref	Reference to the TP Address that is used by the TpNode. This reference is optional in case that the multicast TP Address is used (reference from TpConnection).

Table 6.163: FlexrayArTpNode

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			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*

Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
connectionPrioPdus	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 CommunicationConnector. The target address of the transmitted NPdu is determined by the configured target CommunicationConnector.
flowControlPdu	NPdu	0..1	ref	Please note that this reference is deprecated and will be removed in future. The PDU pool referenced by the FlexRayArTpChannel as nPdu will be used instead. **Tags:** atp.Status=obsolete
multicast	TpAddress	0..1	ref	TP address for 1:n connections.
reversedTpSdu	IPdu	0..1	ref	Reference to the IPdu that is segmented by the Transport Protocol. If support of both sending and receiving is used, this association references the IPdu used for the additional second direction. The source address of the transmitted NPdu is determined by the configured target CommunicationConnector. 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.
transmitPdu	NPdu	*	ref	Please note that this reference is deprecated and will be removed in future. The PDU pool referenced by the FlexRayArTpChannel as nPdu will be used instead. **Tags:** atp.Status=obsolete

Table 6.164: FlexrayArTpConnection

Enumeration	**FrArTpAckType**
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols
Note	Type of Acknowledgement.
Literal	*Description*
ackWithRt	Acknowledgement with retry.
ackWithoutRt	Acknowledgement without retry.
noAck	No acknowledgement.

Table 6.165: FrArTpAckType

Enumeration	**MaximumMessageLengthType**
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols
Note	Type of Acknowledgement.
Literal	***Description***
I4g	SF-E allowed (SF of arbitrary length depending on FrTpPduLength), up to (2**32)-1 byte message length (all FF-x allowed).
iso	Up to (2**12)-1 Byte message length (No FF-Ex or SF-E or AF shall be used and recognized).
iso6	As ISO, but the maximum payload length is limited to 6 byte (SF-I, FF-I, CF). This is necessary to route TP on CAN when using Extended Addressing or Mixed Addressing on CAN.

Table 6.166: MaximumMessageLengthType

6.8.4 CAN Transport Layer

The CAN Transport Layer supports multiple sessions by means of CanTpChannels: Each CanTpChannel uses its own resources, such as internal buffer, timer, state machine and thus can operate independently and simultaneously to other CanTpChannels. The same session can be reused for an arbitrary number of CanTpConnections.

Each CanTpConnection uses its own pair of NPdus: One NPdu, the dataPdu is mandatory for each CanTpConnection, the flowControlPdu is optional depending whether only Single Frames are transferred over the connection.

A CanTpConnection is specifically used for communication between source and target devices. These communication partners are specified using the transmitter and receiver associations to CanTpNode, providing the diagnostic tpAddress and the connection to the topology.

[TPS_SYST_01146] Generic CanTpConnections ⌈ If the transmitter or the receiver of a CanTpConnection is not specified then the CanTpConnection is a generic one (address information is not determined). ⌋

[TPS_SYST_01105] CanTpConnection with several receivers ⌈ In case of several receivers a multicast tpAddress shall be used. ⌋*(RS_SYST_00014)*

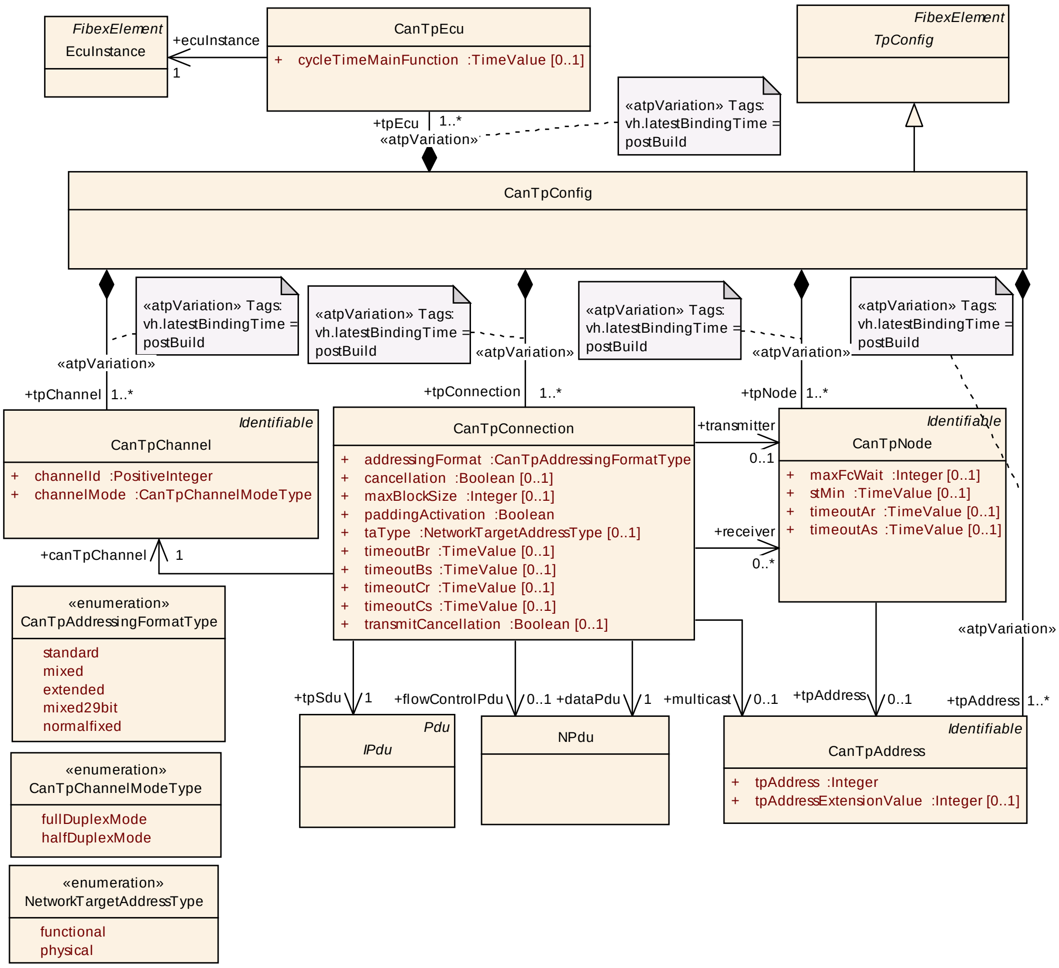

Figure 6.36: CAN Transport Layer Configuration (TransportProtocols: CanTransportProtocol)

The actual payload to be transported by the `CanTpConnection` is specified by the reference `tpSdu` to `IPdu`.

Class	**CanTpConfig**			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	This element defines exactly one CAN TP Configuration. One CanTpConfig element shall be created for each CAN Network in the System. **Tags:** atp.recommendedPackage=TpConfigs			
Base	ARObject,CollectableElement,FibexElement,Identifiable,Multilanguage Referrable,PackageableElement,Referrable,TpConfig			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
tpAddress	CanTpAddress	1..*	aggr	Collection of TP Adresses. atpVariation: Derived, because EcuInstance can vary. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
tpChannel	CanTpChannel	1..*	aggr	Configuration of CAN TP channels. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
tpConnection	CanTpConnection	1..*	aggr	Senders and receivers of CAN TP messages. atpVariation: Derived, because TpNode can vary. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
tpEcu	CanTpEcu	1..*	aggr	Collection of TP Ecus atpVariation: Derived, because EcuInstance can vary. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
tpNode	CanTpNode	1..*	aggr	Senders and receivers of Can TP messages. atpVariation: Derived, because EcuInstance can vary. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild

Table 6.167: CanTpConfig

Class	CanTpChannel			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	Configuration parameters of the CanTp channel.			
Base	ARObject,Identifiable,MultilanguageReferrable,Referrable			
Attribute	Datatype	Mul.	Kind	Note
channelId	PositiveInteger	1	attr	The id of the channel. The value shall be unique for each channel.
channelMode	CanTpChannelModeType	1	attr	The CAN Transport Layer supports half and full duplex channel modes.

Table 6.168: CanTpChannel

Enumeration	CanTpChannelModeType
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols
Note	The CAN Transport Layer supports half and full duplex channel modes.
Literal	Description
fullDuplexMode	full duplex channel mode
halfDuplexMode	half duplex channel mode

Table 6.169: CanTpChannelModeType

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			
Attribute	Datatype	Mul.	Kind	Note
addressingFormat	CanTpAddressingFormatType	1	attr	Declares which communication addressing mode is supported.
canTpChannel	CanTpChannel	1	ref	Reference to the CanTpChannel on which this CanTpConnection is realized.
cancellation	Boolean	0..1	attr	With this switch Tx and Rx Cancellation can be turned on or off.
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

Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
multicast	CanTpAddress	0..1	ref	TP address for 1:n connections.
paddingActivation	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	NetworkTargetAddressType	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.
transmitCancellation	Boolean	0..1	attr	With this switch Transmit Cancellation can be turned on or off for this channel. Please note that this attribute is deprecated and will be removed in future. **Tags:** atp.Status=obsolete; atp.StatusComment=replaced by CanTpConnection.cancellation; atp.StatusRevisionBegin=4.1.3
transmitter	CanTpNode	0..1	ref	The source of the TP connection.

Table 6.170: CanTpConnection

Enumeration	**CanTpAddressingFormatType**
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols
Note	Declares which communication addressing mode is supported.
Literal	*Description*
extended	To use extended addressing format.
mixed	To use mixed 11bit addressing format.
mixed29bit	To use mixed 29bit addressing format
normalfixed	To use normal fixed addressing format
standard	To use normal addressing format.

Table 6.171: CanTpAddressingFormatType

Class	**CanTpAddress**			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	An ECUs TP address on the referenced channel. This represents the diagnostic Address.			
Base	ARObject,Identifiable,MultilanguageReferrable,Referrable			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
tpAddress	Integer	1	attr	An ECUs TP address on the referenced channel. This represents the diagnostic Address.
tpAddress Extension Value	Integer	0..1	attr	If the mixed addressing format is used, this parameter contains the transport protocol address extension value.

Table 6.172: CanTpAddress

Class	**CanTpEcu**			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	ECU specific TP configuration parameters. Each TpEcu element has a reference to exactly one ECUInstance in the topology.			
Base	ARObject			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
cycleTime MainFuncti on	TimeValue	0..1	attr	The period between successive calls to the Main Function of the AUTOSAR TP. Specified in seconds.
ecuInstanc e	EcuInstance	1	ref	Connection to the ECUInstance in the Topology

Table 6.173: CanTpEcu

Class	**CanTpNode**			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	TP Node (Sender or Receiver) provides the TP Address and the connection to the Topology description.			
Base	ARObject,Identifiable,MultilanguageReferrable,Referrable			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*

Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
connector	Communication Connector	0..1	ref	Asscociation to a CommunicationConnector in the topology description. In a System Description this reference is mandatory. In an ECU Extract this reference is optional (references to ECUs that are not part of the ECU Extract shall be avoided).
maxFcWait	Integer	0..1	attr	This attribute defines the maximum number of flow control PDUs that can be consecutively be transmitted by a receiver.
stMin	TimeValue	0..1	attr	Sets the duration of the minimum time the CanTp sender shall wait between the transmissions of two CF N-PDUs.
timeoutAr	TimeValue	0..1	attr	This attribute states the timeout between the PDU transmit request of the Transport Layer to the Can Interface and the corresponding confirmation of the Can Interface on the receiver side (for FC or AF). Specified in seconds.
timeoutAs	TimeValue	0..1	attr	This attribute states the timeout between the PDU transmit request for the first PDU of the group used in the current connection of the Transport Layer to the Can Interface and the corresponding confirmation of the Can Interface (when having sent the last PDU of the group used in this connection) on the sender side (SF-x, FF-x, CF or FC (in case of Transmit Cancellation)). Specified in seconds.
tpAddress	CanTpAddress	0..1	ref	Reference to the TP Address that is used by the TpNode. This reference is optional in case that the multicast TP Address is used (reference from TpConnection).

Table 6.174: CanTpNode

Enumeration	**NetworkTargetAddressType**
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols
Note	Network Target Address type (see ISO 15765-2).
Literal	***Description***
functional	Functional request type
physical	Physical request type

Table 6.175: NetworkTargetAddressType

6.8.5 LIN Transport Layer

LinTpConnection is used for modeling communication resources required for using the LIN Transport Layer. Contrary to the FlexRay and CAN Transport Layers, LIN TP only supports one session per PhysicalChannel.

An arbitrary number of LinTpConnections per LinTpConfig can be defined since the transmission of data from master to slave, using the MasterRequest frame, and the transmission of data from slave to master, using the SlaveResponse frame, needs to be described per NAD the LinMaster uses to address one or more of its LinSlaves.

LinTpConnection uses the dataPdu reference for specifying exactly one NPdu which is to be used for transmitting the data, and it optionally references a flowControl NPdu in order to handle Flow Control Frames if required.

One LinTpConnection is specifically used for communication between one source and one or several target devices. These communication partners are specified using the source and target associations to LinTpNode, providing the diagnostic tpAddress and the connection to the topology. In case of several receivers a multicast tpAddress shall be used.

The actual payload to be transported by the LinTpConnection is specified by the reference linTpNSdu to IPdu.

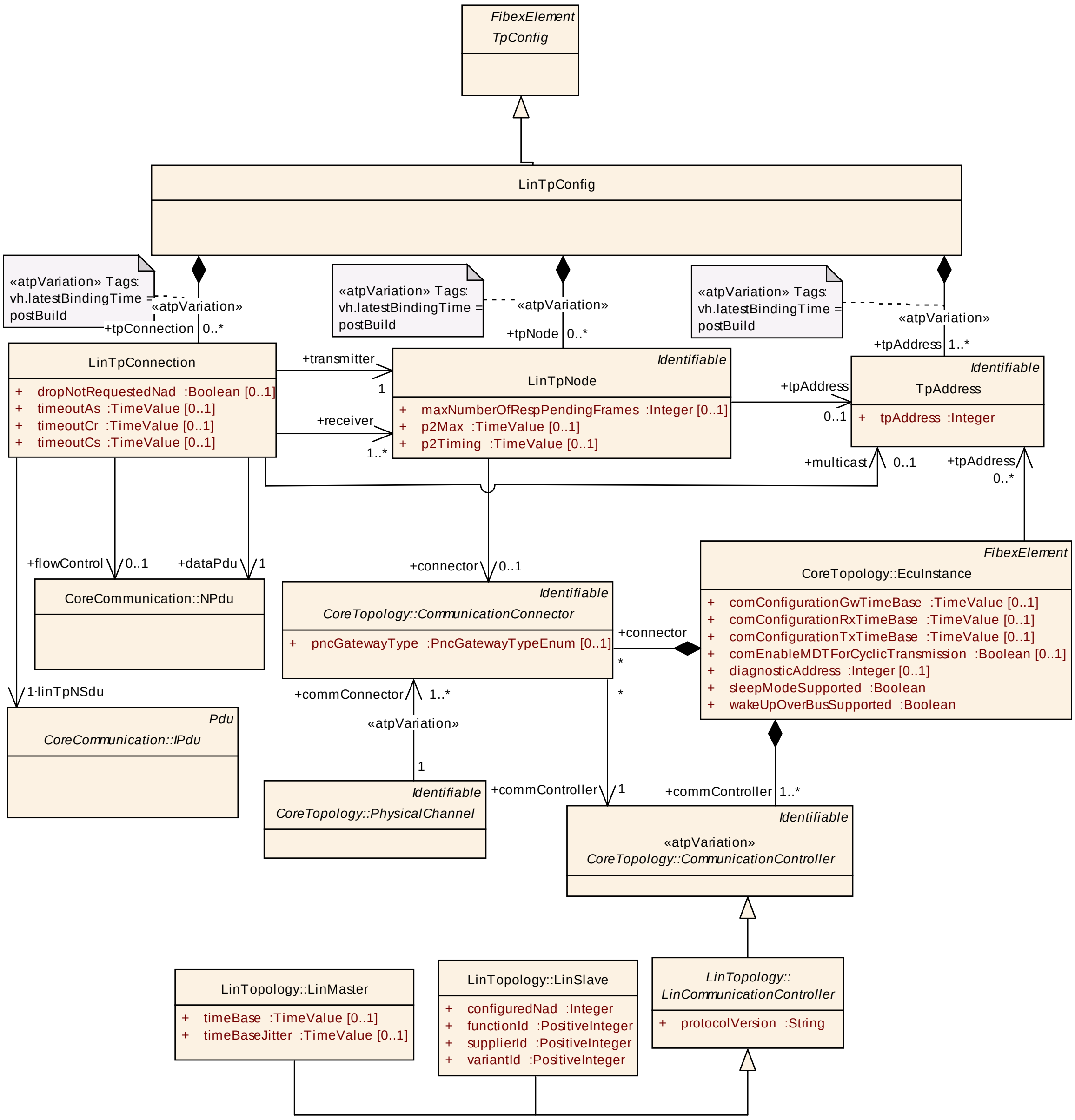

Figure 6.37: LIN Transport Layer Configuration

Class	LinTpConfig			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	This element defines exactly one Lin TP Configuration. One LinTpConfig element shall be created for each Lin Network in the System. **Tags:** atp.recommendedPackage=TpConfigs			
Base	ARObject,CollectableElement,FibexElement,Identifiable,Multilanguage Referrable,PackageableElement,Referrable,TpConfig			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
tpAddress	TpAddress	1..*	aggr	Collection of TpAddresses. atpVariation: Derived, because EcuInstance can vary. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
tpConnection	LinTpConnection	*	aggr	Configuration of LIN TP channels. atpVariation: Derived, because TpNode can vary. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
tpNode	LinTpNode	*	aggr	Senders and receivers of LIN TP messages. atpVariation: Derived, because EcuInstance can vary. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild

Table 6.176: LinTpConfig

Class	LinTpNode			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	TP Node (Sender or Receiver) provides the TP Address and the connection to the Topology description.			
Base	ARObject,Identifiable,MultilanguageReferrable,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
connector	CommunicationConnector	0..1	ref	Asscociation to a CommunicationConnector in the topology description. In a System Description this reference is mandatory. In an ECU Extract this reference is optional (references to ECUs that are not part of the ECU Extract shall be avoided).
maxNumberOfRespPendingFrames	Integer	0..1	attr	Configures the maximum number of allowed response pending frames.

Attribute	Datatype	Mul.	Kind	Note
p2Max	TimeValue	0..1	attr	After reception of a response pending frame the P2 timeout counter is reloaded with the timeout time P2max.
p2Timing	TimeValue	0..1	attr	P2 timeout observation parameter.
tpAddress	TpAddress	0..1	ref	Reference to the TP Address that is used by the TpNode. This reference is optional in case that the multicast TP Address is used (reference from TpConnection).

Table 6.177: LinTpNode

Class	LinTpConnection			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	A LinTP channel represents an internal path for the transmission or reception of a Pdu via LinTp and describes the the sender and the receiver of this particular communication. LinTp supports (per Lin Cluster) the configuration of one Rx Tp-SDU and one Tx Tp-SDU per NAD the LinMaster uses to address one or more of its Lin Slaves. To support this an arbitrary number of LinTpConnections shall be described.			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
dataPdu	NPdu	1	ref	Reference to an NPdu (Single Frame, First Frame or Consecutive Frame). The Single Frame network protocol data unit (SF N_PDU) shall be sent out by the sending network entity and can be received by one or multiple receiving network entities. The Single Frame (SF N_PDU) shall be sent out to transfer a service data unit that can be transferred via a single service request to the data link layer. This network protocol data unit shall be sent to transfer unsegmented messages. The First Frame network protocol data unit (FF N_PDU) identifies the first network protocol data unit (N_PDU) of a segmented message transmitted by a network sending entity and received by a receiving network entity. The Consecutive Frame network protocol data unit (CF N_PDU) transfers segments (N_Data) of the service data unit message data (<MessageData>). All network protocol data units (N_PDUs) transmitted by the sending entity after the First Frame network protocol data unit (FF N_PDU) shall be encoded as Consecutive Frames network protocol data units (CF N_PDUs).
dropNotRequestedNad	Boolean	0..1	attr	Configures if TP Frames of not requested LIN-Slaves are dropped or not.

Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
flowControl	NPdu	0..1	ref	Reference to the Flow Control NPdu. The Flow Control network protocol data unit (FC N_PDU) is identified by the Flow Control protocol control information (FC N_PCI). The Flow Control network protocol data unit (FC N_PDU) instructs a sending network entity to start, stop or resume transmission of CF N_PDUs. The Flow Control network protocol data unit shall be sent by the receiving network layer entity to the sending network layer entity, when ready to receive more data, after correct reception of: a) First Frame network protocol data unit (FF N_PDU) b) the last Consecutive Frame network protocol data unit (CF N_PDU) of a block of Consecutive Frames (CF N_PDU) if further Consecutive Frame network protocol data unit (CF N_PDU) need(s) to be sent.
linTpNSdu	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	LinTpNode	1..*	ref	The target of the TP connection.
timeoutAs	TimeValue	0..1	attr	Time for transmission of the LIN frame (any N-PDU) on the sender side. Specified in seconds.
timeoutCr	TimeValue	0..1	attr	This attribute 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.
transmitter	LinTpNode	1	ref	The source of the TP connection.

Table 6.178: LinTpConnection

6.8.6 SAE J1939 Transport Layer

There are two transport protocol variants defined by J1939: BAM (Broadcast Announce Message), which is a broadcast protocol that does not use any flow control, and CMDT (Connection Mode Data Transfer), which is a point-to-point protocol with flow control and acknowledgment.

BAM uses two NPdus for transport, TP.CM (Transport Protocol Command, flowControlPdu) and TP.DT (Transport Protocol Data, dataPdu). CMDT uses three NPdus, because an additional TP.CM (flowControlPdu) in reverse direction is needed for flow control. The length of TP.CM and TP.DT NPdus is fixed to 8 bytes.

[TPS_SYST_01106] Usage of additional directPdu in case of variable length tpSdu ⌈ In case of variable length tpSdu (with system signals of variable length) an additional directPdu is required:

- it is used if the current length of this tpSdu is up to 8 bytes.
- if the current length of this tpSdu is higher than 8 bytes the tpSdu will be transported via the dataPdu.

⌋*(RS_SYST_00014, RS_SYST_00038)*

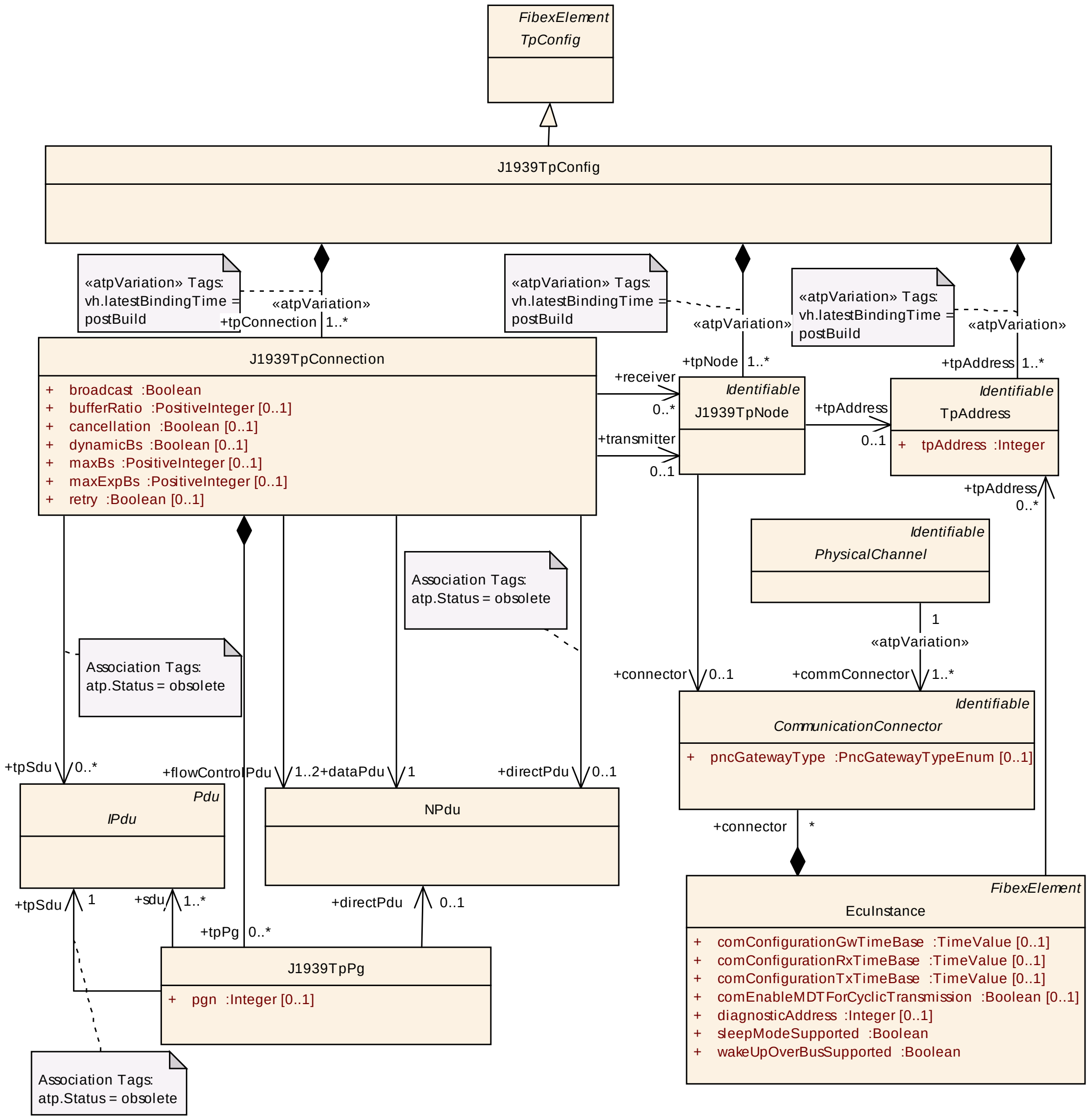

Figure 6.38: J1939 Transport Layer Configuration

A J1939TpConnection is specifically used for communication between source and target devices. These communication partners are specified using the transmitter and receiver associations to J1939TpNode, providing the diagnostic tpAddress and the connection to the topology. BAM (Broadcast Announce Message), is always directed at the target address 0xff, so there is no target address reference necessary for the broadcast situation.

The Parameter Group (PG) to be transported by the J1939TpConnection is specified by the tpPg aggregation.

[TPS_SYST_01147] Generic J1939TpConnections ⌈ If the transmitter or the receiver of a J1939TpConnection is not specified then the J1939TpConnection is a generic one (address information is not determined). ⌋

Class	**J1939TpConfig**			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	This element defines exactly one J1939 TP Configuration. One J1939TpConfig element shall be created for each J1939 Network in the System. **Tags:** atp.recommendedPackage=TpConfigs			
Base	ARObject,CollectableElement,FibexElement,Identifiable,Multilanguage Referrable,PackageableElement,Referrable,TpConfig			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
tpAddress	TpAddress	1..*	aggr	Collection of TP Adresses. atpVariation: Derived, because EcuInstance can vary. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
tpConnection	J1939TpConnection	1..*	aggr	Configuration of J1939 TP connections. atpVariation: Derived, because TpNode can vary. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
tpNode	J1939TpNode	1..*	aggr	Senders and receivers of J1939 TP messages. atpVariation: Derived, because EcuInstance can vary. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild

Table 6.179: J1939TpConfig

Class	J1939TpConnection			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	A J1939TpConnection represents an internal path for the transmission or reception of a Pdu via J1939Tp and describes the the sender and the receiver of this particular communication. The J1939Tp module routes a Pdu (J1939 PGN) through the connection.			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
broadcast	Boolean	1	attr	BAM (Broadcast Announce Message) is a broadcast protocol. If this attribute is set to true broadcast is used. Since address FF is the only broadcast address, there's no reason to configure it.
bufferRatio	PositiveInteger	0..1	attr	Defines usage of available data for dynamic block size calculation when protocol retry is enabled.
cancellation	Boolean	0..1	attr	Enable support for Tx/Rx cancellation.
dataPdu	NPdu	1	ref	Data Message (TP.DT) used by CMDT and BAM. The DataNPdu has a fixed length of 8 bytes.
directPdu	NPdu	0..1	ref	Please note that this reference is deprecated and will be removed in the future. This reference is replaced by the J1939TpPg.directPdu reference. Old description: 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. **Tags:** atp.Status=obsolete
dynamicBs	Boolean	0..1	attr	Enable support for dynamic block size calculation.
flowControlPdu	NPdu	1..2	ref	Reference to the Command NPdus (TP.CM) that are used in the CMDT (Connection Mode Data Transfer) in both directions. BAM uses one TP.CM (Transport Protocol Command). The flowControlNPdu has a fixed length of 8 bytes. Please note that the role name "flowControlIPdu" is misleading and is kept for backward compatibilty reasons.
maxBs	PositiveInteger	0..1	attr	Set maximum block size (number of packets in TP.CM_CTS).
maxExpBs	PositiveInteger	0..1	attr	Set maximum for expected block size (maximum number of packets in TP.CM_RTS).
receiver	J1939TpNode	*	ref	The target of the TP connection.
retry	Boolean	0..1	attr	Enable support for protocol retry.
tpPg	J1939TpPg	*	aggr	J1939 messages (parameter groups, PGs) that can be transferred via this connection.

Attribute	Datatype	Mul.	Kind	Note
tpSdu	IPdu	*	ref	Please note that this reference is deprecated and will be removed in the future. This reference is replaced by the J1939TpPg.tpSdu reference. Old description: Reference to IPdus that are segmented by the Transport Protocol. To support the low-level routing of NPdu's the NPdu is a specialization of an IPdu. More details can be found in the NPdu class description. Nevertheless the J1939TpConnection must not reference a NPdu with this tpSdu reference. **Tags:** atp.Status=obsolete
transmitter	J1939TpNode	0..1	ref	The source of the TP connection.

Table 6.180: J1939TpConnection

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	Datatype	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 CanFrameTriggering with an identifier.
sdu	IPdu	1..*	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.
tpSdu	IPdu	1	ref	Reference to IPdus that are segmented by the Transport Protocol. To support the low-level routing of NPdu's the NPdu is a specialization of an IPdu. More details can be found in the NPdu class description. Nevertheless the J1939TpConnection must not reference a NPdu with this tpSdu reference. **Tags:** atp.Status=obsolete

Table 6.181: J1939TpPg

Class	**J1939TpNode**			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	TP Node (Sender or Receiver) provides the TP Address and the connection to the Topology description.			
Base	ARObject,Identifiable,MultilanguageReferrable,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
connector	CommunicationConnector	0..1	ref	Asscociation to a CommunicationConnector in the topology description. In a System Description this reference is mandatory. In an ECU Extract this reference is optional (references to ECUs that are not part of the ECU Extract shall be avoided).
tpAddress	TpAddress	0..1	ref	Reference to the TP Address that is used by the TpNode. This reference is optional only when no TP is sent and only BAM is received.

Table 6.182: J1939TpNode

6.8.7 Unicast TP Example

The example in Figure 6.39 illustrate the usage of the System Template TP model. In this example the Sender ECU communicates with the Receiver ECU via two Gateways (GW1 and GW2).

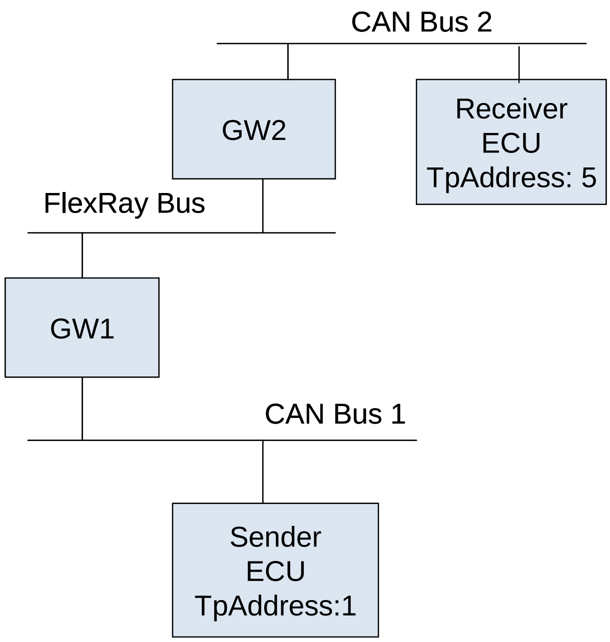

Figure 6.39: TP unicast Example

Modeling in the System Description:

```
CAN Bus 1 (CanTpConfig 1):
CanTpConnection
  transmitter TpNode: Sender ECU, TpAddress: 1
  receiver TpNode: GW1, TpAddress: 5

FlexRay Bus (FlexRayTpConfig):
FlexRayTpConnection
  transmitter TpNode: GW1, TpAddress: 1
  receiver TpNode: GW2, TpAddress: 5

CAN Bus 2 (CanTpConfig 2):
CanTpConnection
  transmitter TpNode: GW2, TpAddress: 1
  receiver TpNode: Receiver ECU, TpAddress: 5
```

Please note that two different `CanTpConfig` elements are created for the two CAN networks. The `TpAddress` of the transmitter `TpNode` is always 1 and the `TpAddress` of the receiver `TpNode` ais always 5, even in the `FlexrayTpConfig` where Gateway ECU1 communicates with Gateway ECU2. The original transmitter and the final receiver are addressed in each connection.

6.8.8 Multicast TP Example

A second example illustrates the usage of the multicast reference.

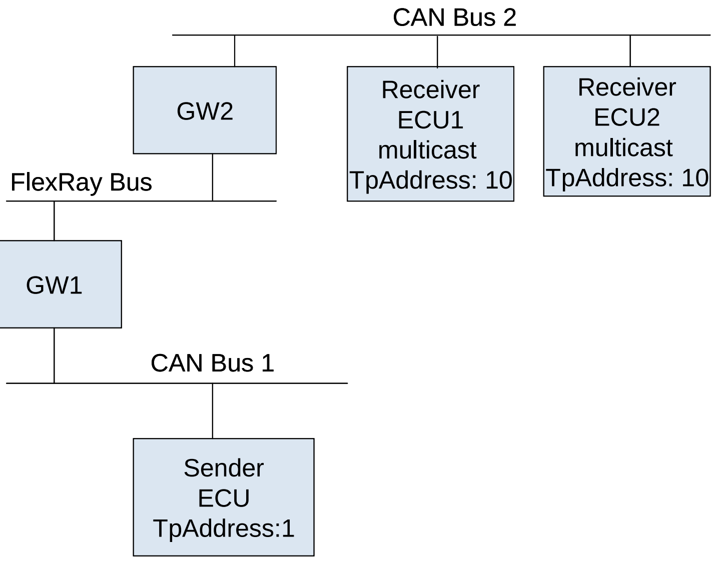

Figure 6.40: TP multicast Example

```
Can Bus 1 (CanTpConfig1):
CanTpConnection
  source TpNode: Sender ECU, TpAddress: 1
  target TpNode: GW1
  multicast TpAddress: 10

FlexRay Bus  (FlexRayTpConfig):
FlexRayTpConnection
  source TpNode: GW1, TpAddress: 1
  target TpNode: GW2
  multicast TpAddress: 10

CAN Bus 2 (CanTpConfig 2):
CanTpConnectionChannel
  source TpNode: GW2, TpAddress: 1
  target TpNode: Receiver ECU1
  target TpNode: Receiver ECU2
  multicast TpAddress: 10
```

Please note that the target `TpNode` does not contain a reference to the `TpAddress`. The multicast `TpAddress` is described by a direct reference from the connection.

6.9 Network Management

The NM specification of AUTOSAR consist of a Generic Network Management Interface Module and of bus specific Network management adaptation layers (CanNm, FrNm, UdpNm, J1939Nm). The AUTOSAR Generic NM Interface module acts as a bus-independent adaptation layer between the bus-specific Network Management modules and the AUTOSAR basic software module Communication Manager. The AUTOSAR Generic NM Interface module is represented by NmCluster, NmEcu, NmCoordinator and NmNode. The bus-specific Network Management attributes are represented by BusspecificNmEcu. See also figure 6.41.

[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. ⌋

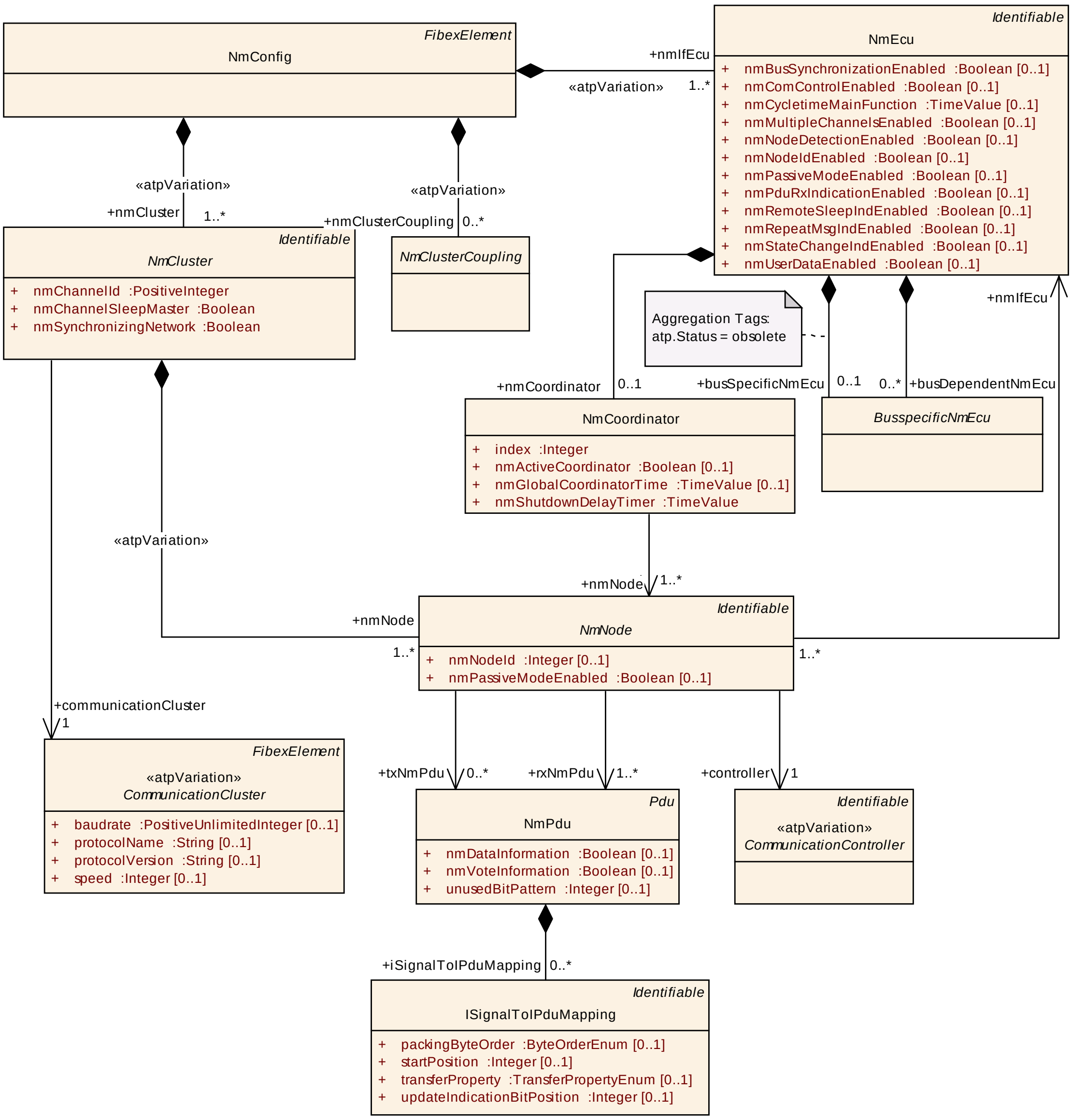

Figure 6.41: Generic Nm elements

The NmCluster contains a set of NmNodes.

The NmNodes are associated with the CommunicationController in the topology and belong to exactly one NmEcu. The reception and transmission of NmPdus is specified with the rxNmPdu and txNmPdu associations to NmPdus.

[TPS_SYST_01107] Definition of NmCoordinator ⌈ An nmCoordinator is connected to two or more CommunicationClusters (via NmNodes) out of which at least two contain the requirement to shutdown synchronously. ⌋

Class	NmConfig			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	Contains the all configuration elements for AUTOSAR Nm. **Tags:** atp.recommendedPackage=NmConfigs			
Base	ARObject,CollectableElement,FibexElement,Identifiable,Multilanguage Referrable,PackageableElement,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
nmCluster	NmCluster	1..*	aggr	Collection of NM Clusters atpVariation: Derived, because cluster can be variable. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
nmCluster Coupling	NmClusterCoup ling	*	aggr	Collection of NmClusterCouplings atpVariation: Derived, because NmCluster can vary. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
nmIfEcu	NmEcu	1..*	aggr	Collection of NM ECUs atpVariation: Derived, because EcuInstance can be variable. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table 6.183: NmConfig

Class	NmCluster (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	Set of NM nodes coordinated with use of the NM algorithm.			
Base	ARObject,Identifiable,MultilanguageReferrable,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
communic ationCluste r	Communication Cluster	1	ref	Asscociation to a CommunicationCluster in the topology description.
nmChanne lId	PositiveInteger	1	attr	Channel identification number of the corresponding channel. Must be unique over all NmClusters.
nmChanne lSleepMast er	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.

Attribute	Datatype	Mul.	Kind	Note
nmNode	NmNode	1..*	aggr	Collection of NmNodes of the NmCluster. atpVariation: Derived, because NmNode can be variable. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
nmSynchronizingNetwork	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 6.184: NmCluster

[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_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). ⌋

Class	NmEcu			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	ECU on which NM is running.			
Base	ARObject,Identifiable,MultilanguageReferrable,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
busDependentNmEcu	BusspecificNmEcu	*	aggr	Cluster specific NmEcu attributes
busSpecificNmEcu	BusspecificNmEcu	0..1	aggr	Busspecific NmEcu attributes. Please note that this aggregation is deprecated and is replaced by the busDependentNmEcu aggregation. **Tags:** atp.Status=obsolete
ecuInstance	EcuInstance	1	ref	Asscociation to an ECUInstance in the topology description.
nmBusSynchronizationEnabled	Boolean	0..1	attr	Enables bus synchronization support.
nmComControlEnabled	Boolean	0..1	attr	Enables the Communication Control support.
nmCoordinator	NmCoordinator	0..1	aggr	Nm ECU may coordinate different clusters.
nmCycletimeMainFunction	TimeValue	0..1	attr	The period between successive calls to the Main Function of the NM Interface in seconds.

Attribute	Datatype	Mul.	Kind	Note
nmMultipleChannelsEnabled	Boolean	0..1	attr	Enables channel multiplicity support. **Tags:** atp.Status=obsolete
nmNodeDetectionEnabled	Boolean	0..1	attr	Enables the Request Repeat Message Request support. Only valid if nmNodeIdEnabled is set to true.
nmNodeIdEnabled	Boolean	0..1	attr	Enables the source node identifier.
nmPassiveModeEnabled	Boolean	0..1	attr	This attribute is deprecated and shall not be used. It is only kept in the model for backward compatibility reasons and will be removed in the future. The passive mode is configurable per channel with the attribute nmPassiveModeEnabled in NmNode. **Tags:** atp.Status=obsolete
nmPduRxIndicationEnabled	Boolean	0..1	attr	Switch for enabling the PDU Rx Indication.
nmRemoteSleepIndEnabled	Boolean	0..1	attr	Switch for enabling remote sleep indication support.
nmRepeatMsgIndEnabled	Boolean	0..1	attr	Switch for enabling the Repeat Message Bit Indication.
nmStateChangeIndEnabled	Boolean	0..1	attr	Enables the CAN Network Management state change notification.
nmUserDataEnabled	Boolean	0..1	attr	Switch for enabling user data support.

Table 6.185: NmEcu

Class	**BusspecificNmEcu (abstract)**			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	Busspecific NmEcu attributes. Please note that this element is deprecated and is replaced by the busDependentNmEcu. **Tags:** atp.Status=obsolete			
Base	ARObject			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
–	–	–	–	–

Table 6.186: BusspecificNmEcu

Class	NmCoordinator			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	A NM coordinator is an ECU, which is connected to at least two busses, and where the requirement exists that shutdown of NM of at least two of these busses (also referred to as coordinated busses) has to be performed synchronously.			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
index	Integer	1	attr	Identification of the NMCoordinator.
nmActiveCoordinator	Boolean	0..1	attr	This attribute indicates whether a NM Coordinator is an active gateway (true) or a passive gateway (false).
nmGlobalCoordinatorTime	TimeValue	0..1	attr	This attribute defines the maximum shutdown time (in seconds) of a connected and coordinated NM-Cluster.
nmNode	NmNode	1..*	ref	reference to busses (via NmNodes) that are coordinated by the NmCoordinator.
nmShutdownDelayTimer	TimeValue	1	attr	This parameter defines the time in seconds which the NM Coordination algorithm shall delay the release of the referenced cluster.

Table 6.187: NmCoordinator

Class	NmNode (abstract)			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	The linking of NmEcus to NmClusters is realized via the NmNodes.			
Base	ARObject,Identifiable,MultilanguageReferrable,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
controller	CommunicationController	1	ref	Asscociation to an CommunicationController in the topology description.
nmIfEcu	NmEcu	1	ref	Reference to the NmEcu that contains this NmNode. (CommunicationController that is referenced by the NmNode 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.
nmPassiveModeEnabled	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 6.188: NmNode

Class	**NmClusterCoupling (abstract)**			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	Attributes that are valid for each of the referenced (coupled) clusters.			
Base	ARObject			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 6.189: NmClusterCoupling

6.9.1 FlexRay Network Management

The following class tables specify the configuration parameters of FlexRay Nm.

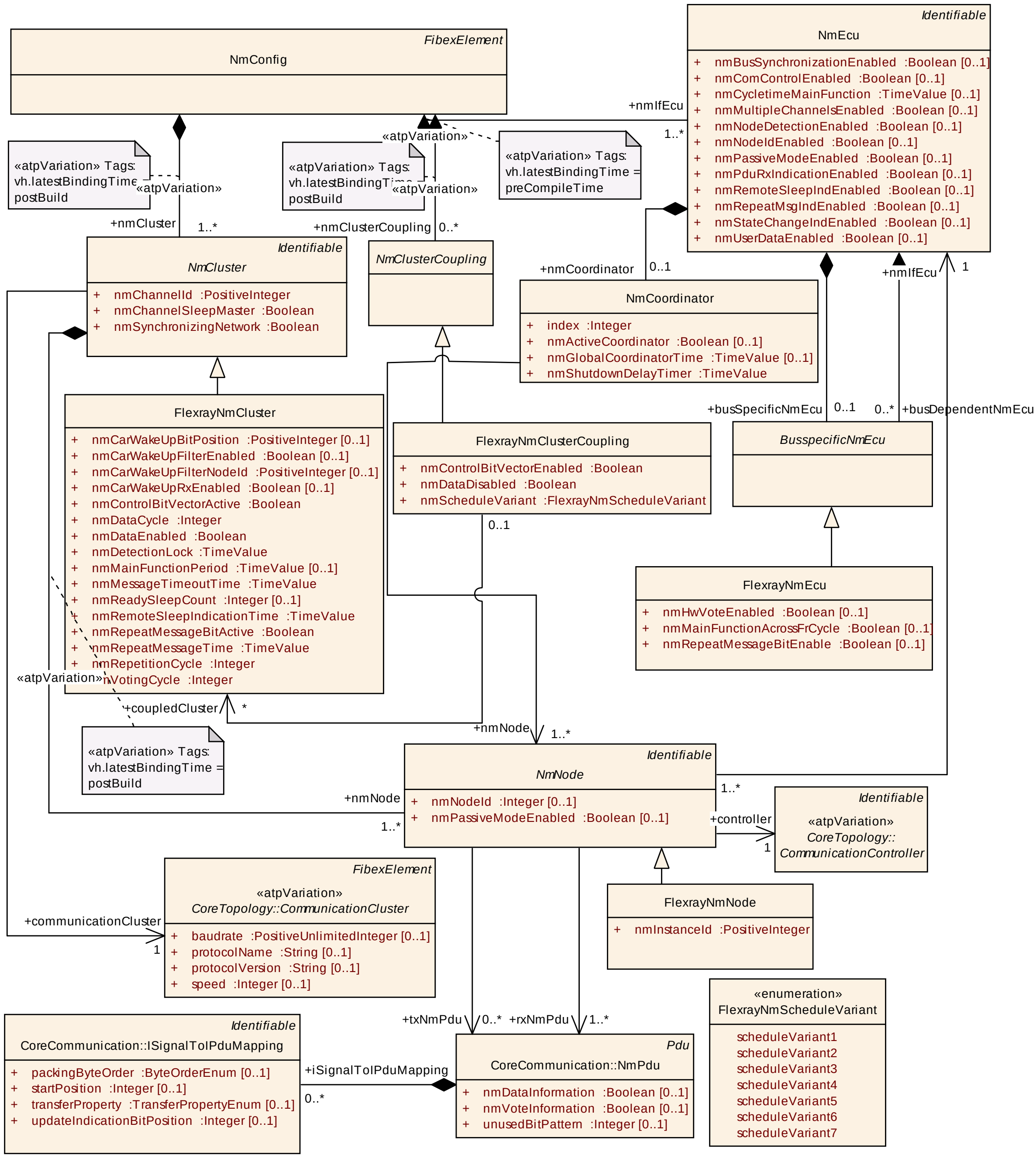

Figure 6.42: FlexRay Network Management Configuration (TransportProtocols: NmFlexRayConfiguration)

Class	**FlexrayNmCluster**			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	FlexRay specific NM cluster attributes.			
Base	ARObject,Identifiable,MultilanguageReferrable,NmCluster,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
nmCarWakeUpBitPosition	PositiveInteger	0..1	attr	Specifies the bit position of the CarWakeUp within the NmPdu.
nmCarWakeUpFilterEnabled	Boolean	0..1	attr	If this attribute is set to true the CareWakeUp filtering is supported. In this case only the CarWakeUp bit within the NmPdu with source node identifier nmCarWakeUpFilterNodeId is considered as CarWakeUp request.
nmCarWakeUpFilterNodeId	PositiveInteger	0..1	attr	Source node identifier for CarWakeUp filtering. If CarWakeUp filtering is supported (nmCarWakeUpFilterEnabled), only the CarWakeUp bit within the NmPdu with source node identifier nmCarWakeUpFilterNodeId is considered as CarWakeUp request.
nmCarWakeUpRxEnabled	Boolean	0..1	attr	If set to true this attribute enables the support of CarWakeUp bit evaluation in received NmPdus.
nmControlBitVectorActive	Boolean	1	attr	Used to activate or deactivate the control bit vector support for a Fr Nm Channel.
nmDataCycle	Integer	1	attr	Number of FlexRay Communication Cycles needed to transmit the Nm Data PDUs of all FlexRay Nm Ecus of this FlexRayNmCluster.
nmDataEnabled	Boolean	1	attr	Switch to enable the separated sending of NM-Data. True: enables False: disables
nmDetectionLock	TimeValue	1	attr	The time for which a node will not set the repeat message request bit even in the presence of a repeat message request (in seconds).
nmMainFunctionPeriod	TimeValue	0..1	attr	Defines the processing cycle of the main function of FrNm module.
nmMessageTimeoutTime	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.

Attribute	Datatype	Mul.	Kind	Note
nmReadySleepCount	Integer	0..1	attr	This attribute is deprecated and will be removed in future. nmReadySleepTime in the FlexrayCommunicationConnector shall be used instead to influence the shutdown behavior of the FlexRay Nm. Old description: Numbers of repetitions in the ready sleep state before NM switches to bus sleep mode. On a value of "1", the NM-State Machine will leave the Ready Sleep State after one NM Repetition Cycle with no "keep awake" votes. **Tags:** atp.Status=obsolete
nmRemoteSleepIndicationTime	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.
nmRepeatMessageBitActive	Boolean	1	attr	Used to activate or deactivate the repeat message bit support for a Fr Nm Channel.
nmRepeatMessageTime	TimeValue	1	attr	Timeout for Repeat Message State in seconds. Defines the time how long the NM shall stay in the Repeat Message State.
nmRepetitionCycle	Integer	1	attr	Number of FlexRay Communication Cycles used to repeat the transmission of the Nm vote Pdus of all FlexRay NmEcus of this FlexRayNmCluster. This value must be an integral multiple of nmVotingCycle.
nmVotingCycle	Integer	1	attr	Number of FlexRay CommunicationCycles needed to transmit the Nm vote of Pdus of all FlexRay NmEcus of this FlexRayNmCluster.

Table 6.190: FlexrayNmCluster

Class	**FlexrayNmEcu**			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	FlexRay specific attributes.			
Base	ARObject,BusspecificNmEcu			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
nmHwVoteEnabled	Boolean	0..1	attr	Switch for enabling the processing of FlexRay Hardware aggregated NM-Votes.
nmMainFunctionAcrossFrCycle	Boolean	0..1	attr	Parameter describing if the execution of the FrNm_Main function crosses theFlexRay cycle boundary or not.
nmRepeatMessageBitEnable	Boolean	0..1	attr	Enables/disables the repeat message bit support

Table 6.191: FlexrayNmEcu

Class	**FlexrayNmClusterCoupling**			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	FlexRay attributes that are valid for each of the referenced (coupled) FlexRay clusters.			
Base	ARObject,NmClusterCoupling			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
coupledCluster	FlexrayNmCluster	*	ref	Reference to coupled FlexRay Clusters.
nmControlBitVectorEnabled	Boolean	1	attr	Enables control bit vector support.
nmDataDisabled	Boolean	1	attr	Disables the transmission of NM-Data.
nmScheduleVariant	FlexrayNmScheduleVariant	1	attr	FrNm schedule variant according to FrNm SWS.

Table 6.192: FlexrayNmClusterCoupling

Class	**FlexrayNmNode**			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	FlexRay specific NM Node attributes.			
Base	ARObject,Identifiable,MultilanguageReferrable,NmNode,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
nmInstanceId	PositiveInteger	1	attr	The NM instance identifier is used for reporting of development errors to DET. It must be unique for each NM instance within one ECU.

Table 6.193: FlexrayNmNode

Enumeration	**FlexrayNmScheduleVariant**
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement
Note	FrNm schedule variant according to FrNm SWS.
Literal	**Description**
scheduleVariant1	NM-Vote and NM Data transmitted within one PDU in static segment. The NM-Vote has to be realized as separate bit within the PDU.
scheduleVariant2	NM-Vote and NM-Data transmitted within one PDU in dynamic segment. The presence (or non-presence) of the PDU corresponds to the NM-Vote
scheduleVariant3	NM-Vote and NM-Data are transmitted in the static segment in separate PDUs. This alternative is not recommended => Alternative 1 should be used instead.
scheduleVariant4	NM-Vote transmitted in static and NM-Data transmitted in dynamic segment.
scheduleVariant5	NM-Vote is transmitted in dynamic and NM-Data is transmitted in static segment. This alternative is not recommended => Variants 2 or 6 should be used instead.
scheduleVariant6	NM-Vote and NM-Data are transmitted in dynamic segment in separate PDUs.
scheduleVariant7	NM-Vote and a copy of the CBV are transmitted in the static segment (using the FlexRay NM Vector support) and NM-Data is transmitted in the dynamic segment

Table 6.194: FlexrayNmScheduleVariant

6.9.2 CAN Network Management

The following class tables specify the configuration parameters of CAN Nm.

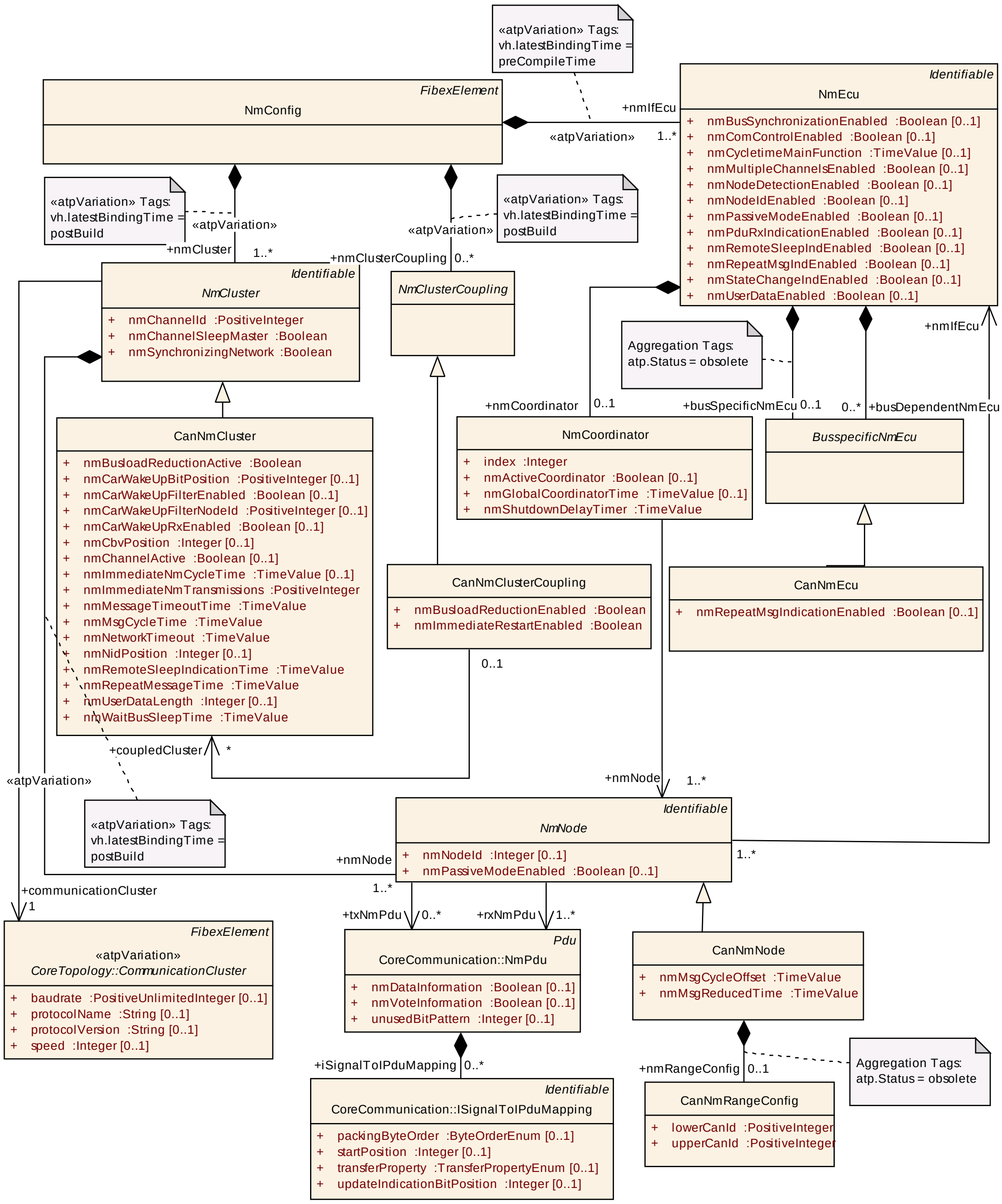

Figure 6.43: CAN Network Management Configuration (TransportProtocols: NmCanConfiguration)

Class	**CanNmCluster**			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	Can specific NmCluster attributes			
Base	ARObject,Identifiable,MultilanguageReferrable,NmCluster,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
nmBusloadReductionActive	Boolean	1	attr	It determines if bus load reduction for the respective CanNm channel is active or not.
nmCarWakeUpBitPosition	PositiveInteger	0..1	attr	Specifies the bit position of the CarWakeUp within the NmPdu.
nmCarWakeUpFilterEnabled	Boolean	0..1	attr	If this attribute is set to true the CareWakeUp filtering is supported. In this case only the CarWakeUp bit within the NmPdu with source node identifier nmCarWakeUpFilterNodeId is considered as CarWakeUp request.
nmCarWakeUpFilterNodeId	PositiveInteger	0..1	attr	Source node identifier for CarWakeUp filtering. If CarWakeUp filtering is supported (nmCarWakeUpFilterEnabled), only the CarWakeUp bit within the NmPdu with source node identifier nmCarWakeUpFilterNodeId is considered as CarWakeUp request.
nmCarWakeUpRxEnabled	Boolean	0..1	attr	If set to true this attribute enables the support of CarWakeUp bit evaluation in received NmPdus.
nmCbvPosition	Integer	0..1	attr	Defines the position of the control bit vector within the NmPdu (Byte positon). If this attribute is not configured, the Control Bit Vector is not used.
nmChannelActive	Boolean	0..1	attr	Please note that this attribute is deprecated and will be removed in future. This switch determines if the respective CanNm channel is active or not. Indicates whether a particular CanNm 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. **Tags:** atp.Status=obsolete; atp.StatusRevisionBegin=4.1.2
nmImmediateNmCycleTime	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 CanNmImmediateNmTransmissions is greater one.
nmImmediateNmTransmissions	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.

Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
nmMessageTimeoutTime	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.
nmMsgCycleTime	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.
nmNetworkTimeout	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.
nmRemoteSleepIndicationTime	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.
nmRepeatMessageTime	TimeValue	1	attr	Timeout for Repeat Message State in seconds. Defines the time how long the NM shall stay in the Repeat Message State.
nmUserDataLength	Integer	0..1	attr	Defines the length of the user data contained in the NmPdu. Please note that this attribute is deprecated and will be removed in future. **Tags:** atp.Status=obsolete; atp.StatusRevisionBegin=4.1.3
nmWaitBusSleepTime	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 6.195: CanNmCluster

[constr_3069] Allowed `CanNmCluster.nmNidPosition` values ⌈ The value of `CanNmCluster.nmNidPosition` shall only be set to either bit 0 (byte 0) or bit 8 (byte 1). ⌋

[constr_3070] Allowed `CanNmCluster.nmCbvPosition` values ⌈ The value of `CanNmCluster.nmCbvPosition` shall only be set to either bit 0 (byte 0) or bit 8 (byte 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. ⌋

Class	CanNmEcu			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	CAN specific attributes.			
Base	ARObject,BusspecificNmEcu			
Attribute	Datatype	Mul.	Kind	Note
nmRepeatMsgIndicationEnabled	Boolean	0..1	attr	Enable/disable the notification that a RepeatMessageRequest bit has been received. This attribute is deprecated and shall be not used. It will be removed in the future. The nmRepeatMsgIndEnabled attribute in NmEcu shall be used instead. **Tags:** atp.Status=obsolete

Table 6.196: CanNmEcu

Class	CanNmClusterCoupling			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	CAN attributes that are valid for each of the referenced (coupled) CAN clusters.			
Base	ARObject,NmClusterCoupling			
Attribute	Datatype	Mul.	Kind	Note
coupledCluster	CanNmCluster	*	ref	Reference to coupled CAN Clusters.
nmBusloadReductionEnabled	Boolean	1	attr	Enables busload reduction support
nmImmediateRestartEnabled	Boolean	1	attr	Enables the asynchronous transmission of a CanNm PDU upon bus-communication request in Prepare-Bus-Sleep mode.

Table 6.197: CanNmClusterCoupling

Class	CanNmNode			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	CAN specific NM Node attributes.			
Base	ARObject,Identifiable,MultilanguageReferrable,NmNode,Referrable			
Attribute	Datatype	Mul.	Kind	Note
nmMsgCycleOffset	TimeValue	1	attr	Node specific time offset in the periodic transmission node. It determines the start delay of the transmission. Specified in seconds.
nmMsgReducedTime	TimeValue	1	attr	Node specific bus cycle time in the periodic transmission mode with bus load reduction. Specified in seconds.

Attribute	Datatype	Mul.	Kind	Note
nmRangeConfig	CanNmRangeConfig	0..1	aggr	Defines the CANID ranges that are used for Nm. This range definition is redundant to the attribute "rxIdentifierRange" of CanFrameTriggering. For backward compatibility reasons this redundancy shall be preserved and both shall be defined. In future this element will be removed from the model. **Tags:** atp.Status=obsolete

Table 6.198: CanNmNode

Class	CanNmRangeConfig			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	Defines the CANID ranges that are used for Nm. This range definition is redundant to the attribute "rxIdentifierRange" of CanFrameTriggering. For backward compatibility reasons this redundancy shall be preserved and both shall be defined. In future this element will be removed from the model. **Tags:** atp.Status=obsolete			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
lowerCanId	PositiveInteger	1	attr	Lower CAN Identifier of a receive CAN L-PDU for identifier range definition.
upperCanId	PositiveInteger	1	attr	Upper CAN Identifier of a receive CAN L-PDU for identifier range definition.

Table 6.199: CanNmRangeConfig

6.9.3 LIN Network Management

No relevant system information is described in the LinNm configuration. In AUTOSAR there is no communication between LinNm and LinIf and there are no dedicated LinNm frames. Therefore a LinNm model in the System Template is unnecessary.

6.9.4 UDP Network Management

The following class tables specify the configuration parameters of UDP Nm.

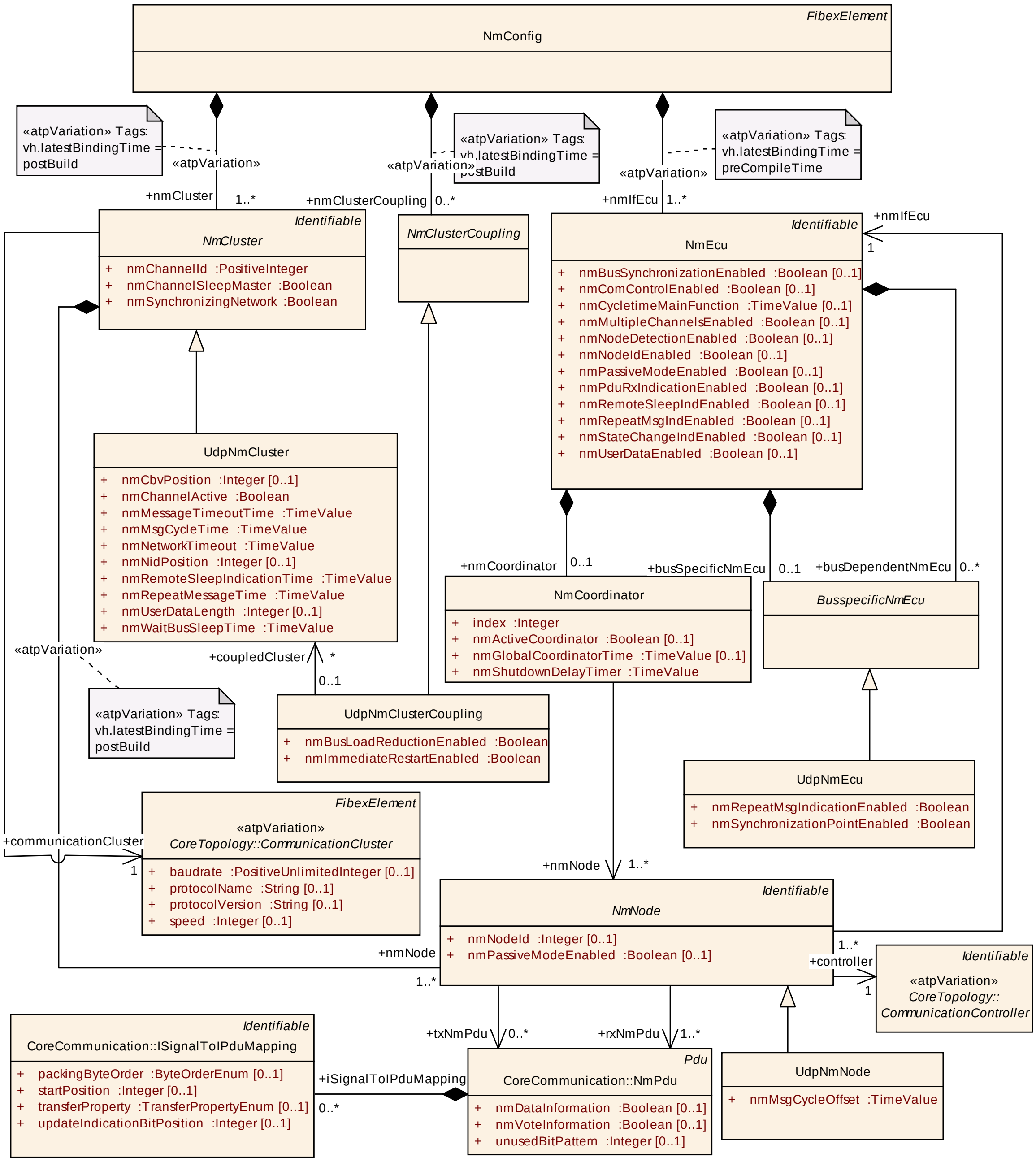

Figure 6.44: UDP Network Management Configuration (TransportProtocols: NmUdpConfiguration)

Class	UdpNmCluster			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	Udp specific NmCluster attributes			
Base	ARObject,Identifiable,MultilanguageReferrable,NmCluster,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
nmCbvPosition	Integer	0..1	attr	Defines the position of the control bit vector within the NmPdu (Byte positon).
nmChannelActive	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.
nmMessageTimeoutTime	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.
nmMsgCycleTime	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.
nmNetworkTimeout	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.
nmRemoteSleepIndicationTime	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.
nmRepeatMessageTime	TimeValue	1	attr	Timeout for Repeat Message State in seconds. Defines the time how long the NM shall stay in the Repeat Message State.
nmUserDataLength	Integer	0..1	attr	Defines the length of the user data contained in the NmPdu. Please note that this attribute is deprecated and will be removed in future. **Tags:** atp.Status=obsolete; atp.StatusRevisionBegin=4.1.3
nmWaitBusSleepTime	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 6.200: UdpNmCluster

[constr_3078] Allowed `UdpNmCluster.nmNidPosition` values ⌈ The value of `UdpNmCluster.nmNidPosition` shall only be set to either bit 0 (byte 0) or bit 8 (byte 1). ⌋

[constr_3079] Allowed `UdpNmCluster.nmCbvPosition` values ⌈ The value of `UdpNmCluster.nmCbvPosition` shall only be set to either bit 0 (byte 0) or bit 8 (byte 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. ⌋

Class	**UdpNmEcu**			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	Udp NM specific ECU attributes.			
Base	ARObject,BusspecificNmEcu			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
nmRepeatMsgIndicationEnabled	Boolean	1	attr	Enable/disable the notification that a RepeatMessageRequest bit has been received.
nmSynchronizationPointEnabled	Boolean	1	attr	Enable/disable the NM Coordination algorithm to being able to initiate the synchronization algorithm.

Table 6.201: UdpNmEcu

Class	**UdpNmClusterCoupling**			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	Udp attributes that are valid for each of the referenced (coupled) UdpNm clusters.			
Base	ARObject,NmClusterCoupling			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
coupledCluster	UdpNmCluster	*	ref	Reference to coupled UdpNm Clusters.
nmBusLoadReductionEnabled	Boolean	1	attr	Enables busload reduction support
nmImmediateRestartEnabled	Boolean	1	attr	Enables the asynchronous transmission of a CanNm PDU upon bus-communication request in Prepare-Bus-Sleep mode.

Table 6.202: UdpNmClusterCoupling

Class	**UdpNmNode**			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	Udp specific NM Node attributes.			
Base	ARObject,Identifiable,MultilanguageReferrable,NmNode,Referrable			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
nmMsgCycleOffset	TimeValue	1	attr	Node specific time offset in the periodic transmission node. It determines the start delay of the transmission. Specified in seconds.

Table 6.203: UdpNmNode

6.9.5 J1939 Network Management

The following class tables specify the configuration parameters of J1939 Nm.

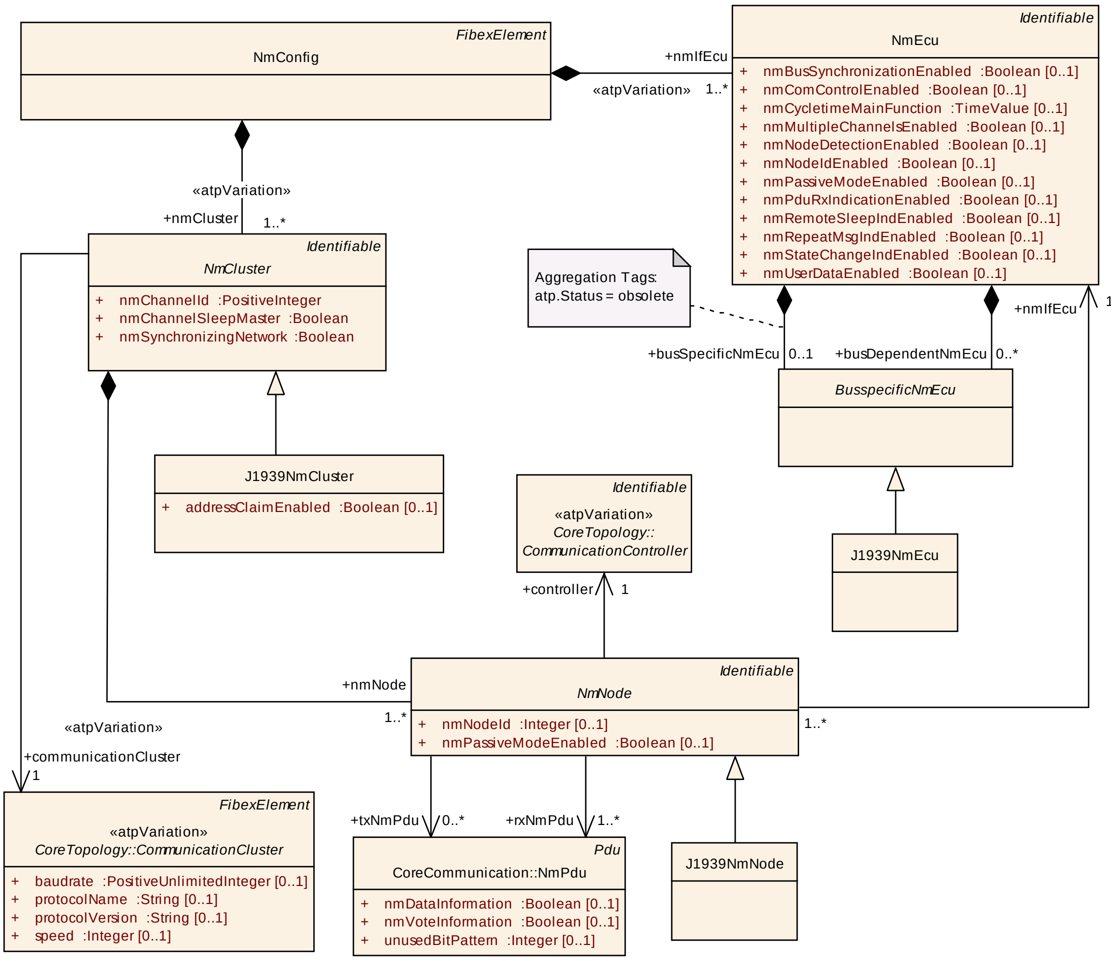

Figure 6.45: J1939 Network Management Configuration (TransportProtocols: NmJ1939Configuration)

Class	J1939NmCluster			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	J1939 specific NmCluster attributes			
Base	ARObject,Identifiable,MultilanguageReferrable,NmCluster,Referrable			
Attribute	Datatype	Mul.	Kind	Note
addressClaimEnabled	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 6.204: J1939NmCluster

Class	J1939NmNode			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	J1939 specific NM Node attributes.			
Base	ARObject,Identifiable,MultilanguageReferrable,NmNode,Referrable			
Attribute	Datatype	Mul.	Kind	Note
–	–	–	–	–

Table 6.205: J1939NmNode

Class	J1939NmEcu			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	J1939 NmEcu specific attributes.			
Base	ARObject,BusspecificNmEcu			
Attribute	Datatype	Mul.	Kind	Note
–	–	–	–	–

Table 6.206: J1939NmEcu

6.10 Fan-out

AUTOSAR supports three different fan-outs:

- Signal fan-out
- Pdu fan-out
- Frame fan-out

6.10.1 Signal fan-out

A Signal fan-out can either be RTE fan-out or COM Signal Gateway fan-out. The details are explained in the following subchapters.

6.10.1.1 RTE fan-out

The RTE supports a "signal fan-out" where one SystemSignal is sent in several IPdus.

[TPS_SYST_01109] RTE fan-out support ⌈ The RTE fan-out (signal fan-out) is described by the relation between SystemSignal and ISignal.

In the case of a "signal fan-out", several ISignals refer to the same SystemSignal (see example in Figure 6.46). ⌋

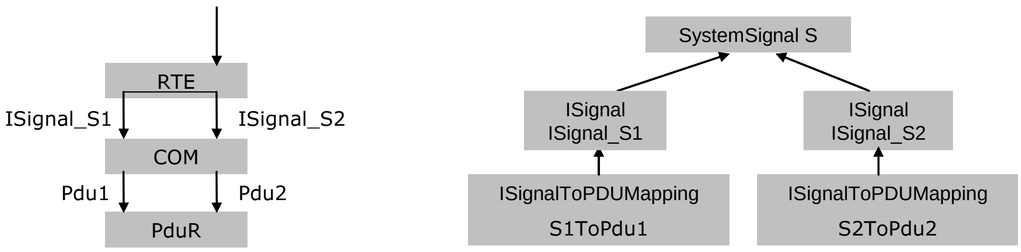

Figure 6.46: RTE fan-out

6.10.1.2 COM Signal Gateway fan-out

In Com [19] the Signal Gateway supports a fan-out where an incoming signal is routed to several destinations.

[TPS_SYST_01110] Com Signal Gateway fan-out support ⌈ A Signal Gateway fan-out (1:n routing) is described with the definition of several ISignalMappings in the Gateway description, which all refer to the same source ISignalTriggering. ⌋

Note that [constr_3514] applies for the relation between ISignalToIPduMapping to ISignal.

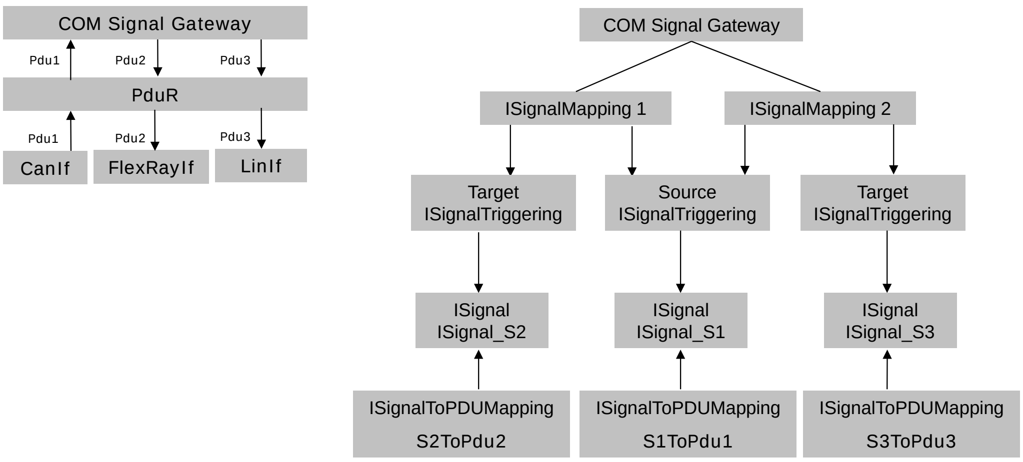

Figure 6.47: Com Gateway fan-out

6.10.2 Pdu fan-out

6.10.2.1 Pdu Router fan-out

The `Pdu Router` supports the "PDU fan-out" where one `IPdu` is sent to multiple destinations.

[TPS_SYST_01111] Pdu Router fan-out support ⌈ The `Pdu Router` fan-out is described by several `PduTriggering` elements pointing to the same `Pdu`[5].

The sending ECU/PDU router has an output `IPduPort` that has the value of `communicationDirection` set to `out` and is referenced by the `PduTriggering`. According to the Cluster/Channel aggregation, the `Pdu Router` determines the clusters to use in its routing. ⌋

[TPS_SYST_01112] `FlexrayCluster` Pdu Router interaction ⌈ The following condition applies only in case of FlexRay on the same `FlexrayCluster` if two `PduTriggering`s refer to the same `Pdu`: this `Pdu` shall only be sent once to the FlexRay Interface. In other words the Pdu Router sends only one Pdu Transmission request to the FlexRay Interface. ⌋

[5]Chapter 6.1 defines which `Pdu` types are routed by the `Pdu Router`

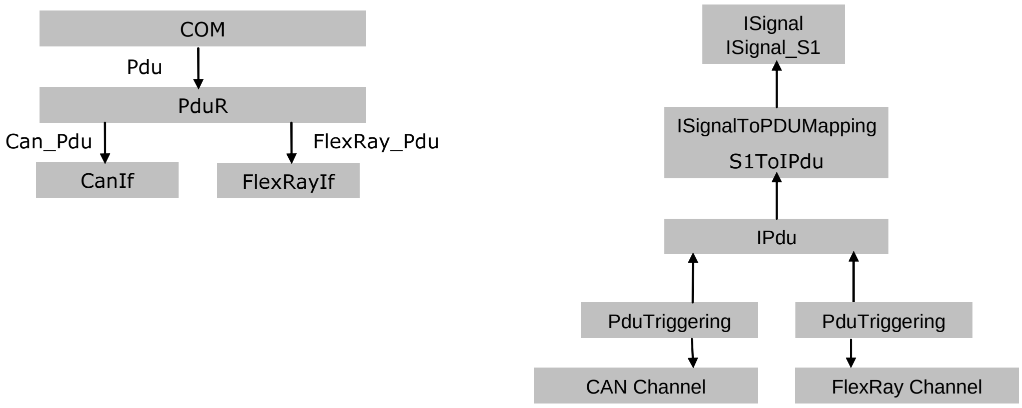

Figure 6.48: Pdu Router fan-out

6.10.2.2 Flexray Interface fan-out

The Flexray interface supports a fan-out where one `Pdu` is mapped into more than one frame on the same `CommunicationCluster`.

[TPS_SYST_01113] FlexRay Interface fan-out support ⌈ The fan-out done in the FlexRay Interface is described by the `FrameTriggering` element (e.g. the same Frame with the same Pdu content is transmitted over FlexRay channel A and FlexRay channel B, see example in Figure 6.49). ⌋

If the fan-out is specified between different FlexRay channels of the same cluster it shall be handled by the FlexRay Interface.

The Flexray Interface does NOT handle fan-out/in between different clusters.

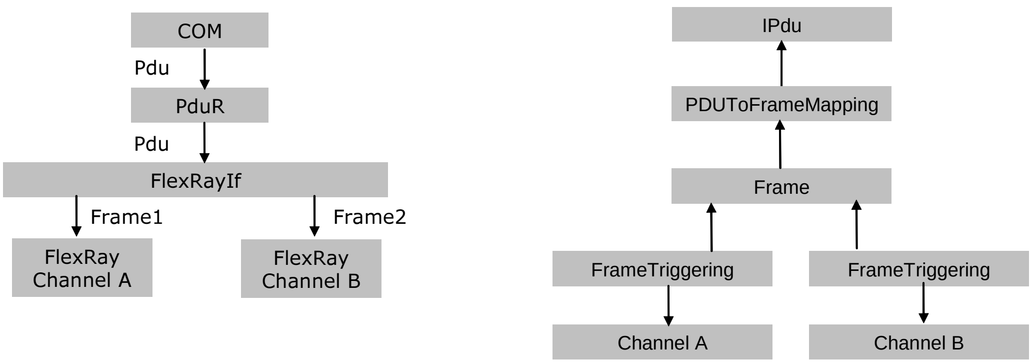

Figure 6.49: Bus Interface fan-out

6.10.3 Frame fan-out

[TPS_SYST_01114] Frame fan-out support ⌈ For the same `Frame`, if several `FrameTriggering`s with the same direction exist on more than one `PhysicalChan-`

nel of the same CommunicationCluster the fan-out/in is handled by the interface.
⌋

6.11 Support of Complex Drivers

The System Template allows the integration of custom communication means into AUTOSAR EcuInstances.

[TPS_SYST_01115] CDD communication support ⌈ The elements UserDefinedPdu and UserDefinedIPdu shall be used to describe the Pdu-based communication via Complex Drivers. ⌋*(RS_SYST_00043)*

The UserDefinedPdu and UserDefinedIPdu elements are described in chapter 6.3 in more detail.

The UserDefinedIPdu can be used to describe the communication if a new BSW module was added above the PduR, e.g a Diagnostic Service.

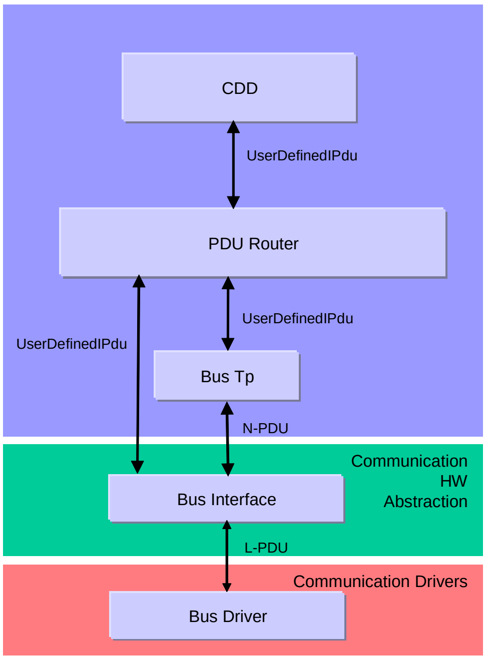

Figure 6.50: CDD over PduR

The UserDefinedPdu can be used to describe the communication if a new BSW module was added above an Interface, e.g. a new Nm module or XCP.

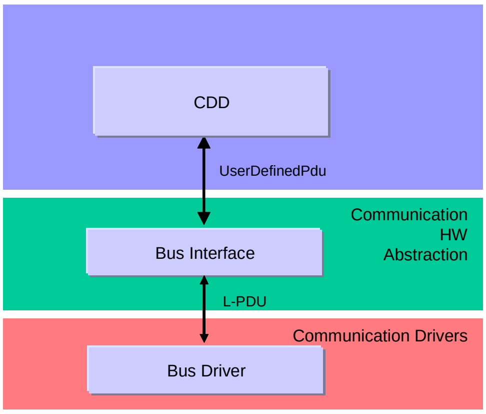

Figure 6.51: CDD over Bus Interface

7 Gateways

A gateway is a function within an EcuInstance that performs as a FrameMapping, IPduMapping or ISignalMapping function between two or more CommunicationClusters.

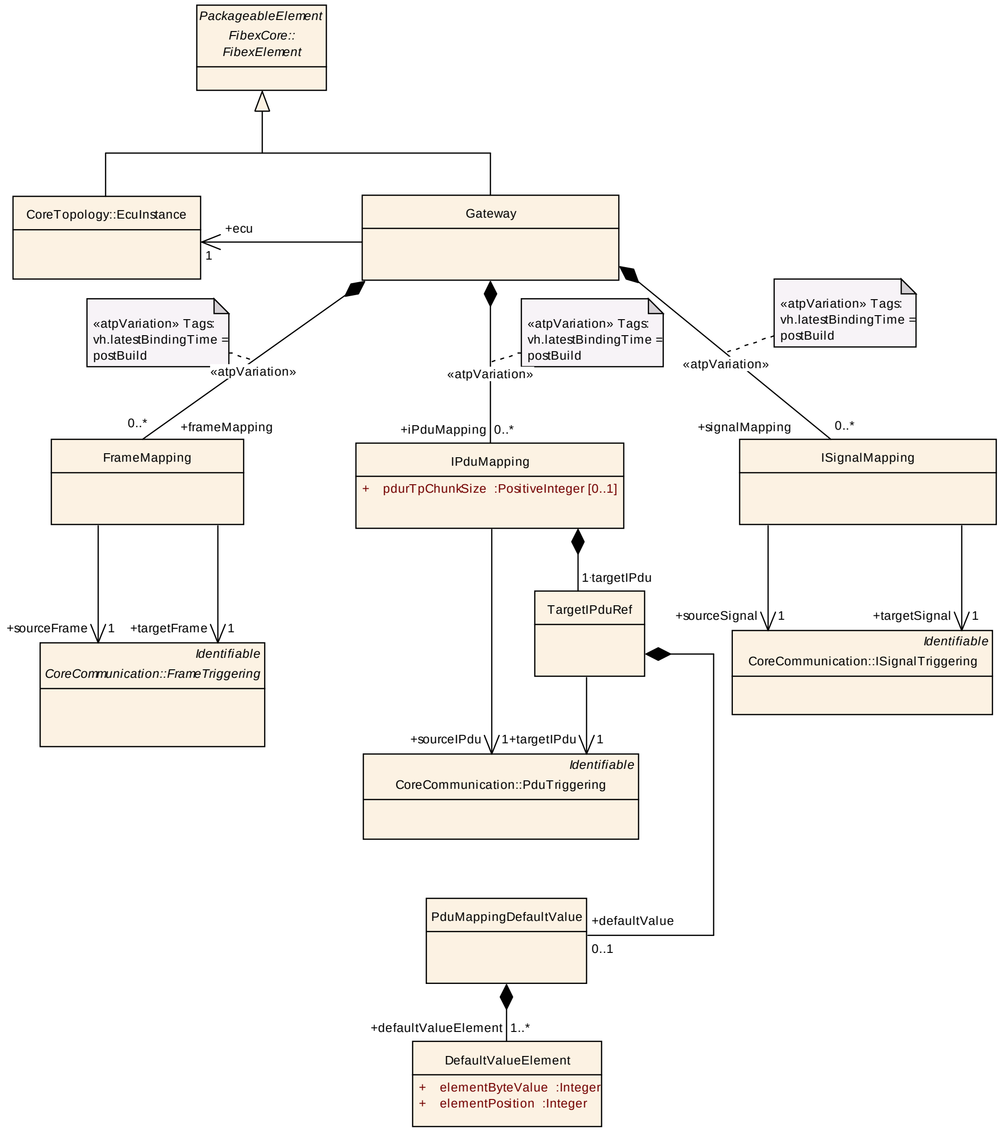

Figure 7.1: Communication Overview (Fibex4Multiplatform: Gateway)

Figure 7.1 shows the meta-model for the Gateway description in the System Template.

Class	**Gateway**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Multiplatform			
Note	A gateway is an ECU that is connected to two or more clusters (channels, but not redundant), and performs a frame, Pdu or signal mapping between them. **Tags:** atp.recommendedPackage=Gateways			
Base	ARObject,CollectableElement,FibexElement,Identifiable,Multilanguage Referrable,PackageableElement,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
ecu	EcuInstance	1	ref	Reference to one ECU instance that implements the gateway.
frameMapping	FrameMapping	*	aggr	Frame Gateway: The entire source frame is mapped as it is onto the target frame (what in general is only possible inside of a common platform). In this case source and target frame should be the identical object. atpVariation: If frames are variable in clusters, the gateway frame mapping needs to be variable, too. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
iPduMapping	IPduMapping	*	aggr	IPdu Gateway: Arranges those IPdus that are transferred by the gateway from one channel to the other in pairs and defines the mapping between them. atpVariation: If PDUs are variable in clusters, the gateway PDU mapping needs to be variable, too. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
signalMapping	ISignalMapping	*	aggr	Signal Gateway: Arranges those signals that are transferred by the gateway from one channel to the other in pairs and defines the mapping between them. atpVariation: If signals are variable in clusters, the gateway signal mapping needs to be variable, too. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild

Table 7.1: Gateway

7.1 Frame Mapping

The FrameMapping arranges those FrameTriggerings that are transferred by the Gateway from one PhysicalChannel to the other in pairs and defines the mapping between them. Each pair consists of a sourceFrame and a targetFrame referencing to a FrameTriggering.

[TPS_SYST_01116] Frame Mapping is not supported by the AUTOSAR BSW ⌈
The FrameMapping is not supported by the AUTOSAR BSW. ⌋

The existence is optional and has been incorporated into the System Template mainly for compatibility in order to allow interchange between FIBEX and AUTOSAR descriptions.

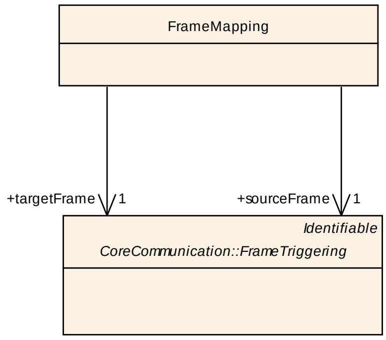

Figure 7.2: Frame Mapping (Fibex4Multiplatform: FrameMapping)

Class	**FrameMapping**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Multiplatform			
Note	The entire source frame is mapped as it is onto the target frame (what in general is only possible inside of a common platform). In this case source and target frame should be the identical object. Each pair consists in a SOURCE and a TARGET referencing to a FrameTriggering. The Frame Mapping is not supported by the Autosar BSW. The existence is optional and has been incorporated into the System Template mainly for compatibility in order to allow interchange between FIBEX and AUTOSAR descriptions.			
Base	ARObject			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
introduction	DocumentationBlock	0..1	aggr	This represents introductory documentation about the frame mapping.
sourceFrame	FrameTriggering	1	ref	Source destination of the referencing mapping.
targetFrame	FrameTriggering	1	ref	Target destination of the referencing mapping.

Table 7.2: FrameMapping

7.2 IPdu Mapping

[TPS_SYST_01117] Pdu Gateway support ⌈ The IPduMapping arranges those IPdus that are transferred by the Gateway from one PhysicalChannel to the other (or the same) PhysicalChannel in pairs and defines the mapping between them. Each pair consist of a sourceIpdu and a targetIpdu referencing to a PduTriggering. ⌋

For FlexRay: If a Pdu is gatewayed to more than one PhysicalChannel of the same CommunicationCluster, all of this gateway relationships shall be specified. Therefore, all affected PduTriggerings shall be referenced in the gateway mappings.

[TPS_SYST_01118] Support of Multicast Pdu routing ⌈ The 1:n multicast routing is supported with the definition of several IPduMappings. ⌋

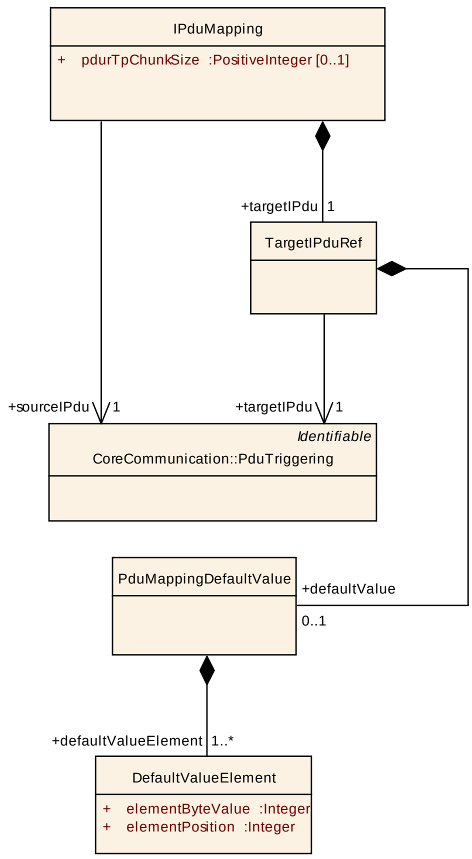

Figure 7.3: I-Pdu Mapping (Fibex4Multiplatform: IPduMapping)

Class	**IPduMapping**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Multiplatform			
Note	Arranges those IPdus that are transferred by the gateway from one channel to the other in pairs and defines the mapping between them.			
Base	ARObject			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
introduction	DocumentationBlock	0..1	aggr	This represents introductory documentation about the IPdu mapping.
pdurTpChunkSize	PositiveInteger	0..1	attr	Optionally defines the to be configured Pdu Router TpChunkSize for this routing relation.
sourceIPdu	PduTriggering	1	ref	Source destination of the referencing mapping.
targetIPdu	TargetIPduRef	1	aggr	Target destination of the referencing mapping.

Table 7.3: IPduMapping

Class	**TargetIPduRef**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Multiplatform			
Note	Target destination of the referencing mapping.			
Base	ARObject			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
defaultValue	PduMappingDefaultValue	0..1	aggr	If no I-Pdu has been received a default value will be distributed.
targetIPdu	PduTriggering	1	ref	IPdu Reference

Table 7.4: TargetIPduRef

Class	**PduMappingDefaultValue**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Multiplatform			
Note	Default Value which will be distributed if no I-Pdu has been received since last sending.			
Base	ARObject			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
defaultValueElement	DefaultValueElement	1..*	aggr	The default value consists of a number of elements. Each default value element is represented by the element and the position in an array.

Table 7.5: PduMappingDefaultValue

Class	**DefaultValueElement**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Multiplatform			
Note	The default value consists of a number of elements. Each element is one byte long and the number of elements is specified by SduLength.			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
elementByteValue	Integer	1	attr	The integer value of a freely defined data byte.
elementPosition	Integer	1	attr	This attribute specifies the byte position of the element within the default value

Table 7.6: DefaultValueElement

7.3 Signal Mapping

[TPS_SYST_01119] Signal Gateway support ⌈ The ISignalMapping defines the mapping between ISignals and ISignalGroups that are transferred by the Gateway from one PhysicalChannel to the other (or the same) PhysicalChannel. Each mapping pair consists of a sourceSignal and a targetSignal referencing an ISignalTriggering. Each ISignalTriggering points to either an ISignal or an ISignalGroup. The ISignal refers to the to be routed SystemSignal, the ISignalGroup refers to the to be routed SystemSignalGroup. ⌋

[constr_3051] Restriction of ISignalMapping references ⌈ If the sourceSignal references an ISignal then the targetSignal shall also reference an ISignal. ⌋

[TPS_SYST_01155] Routing of ISignalGroups ⌈ If the sourceSignal references an ISignalGroup then the targetSignal can reference either an ISignalGroup or an ISignal. ⌋

[constr_3052] Complete ISignalMapping of ISignalGroup signals ⌈ If an ISignalMapping to an ISignal that is a member of a ISignalGroup exists then an ISignalMapping to the enclosing ISignalGroup shall exist as well. ⌋

The routing of ISignalGroups is specified by defining the routing of the ISignalGroup pointing to the SystemSignalGroup (signal group). When performing a signal group routing the pairing of the ISignals is done by the ISignal reference from ISignalGroup to ISignal.

By default, the shortNames of ISignal elements are used to identify the matching elements of ISignalGroups referenced in the scope of ISignalMapping. In case of non-matching shortName it is required to explicitly specify which ISignals of ISignalGroups shall correlate to each other's.

This definition is provided by means of the explicit definition of ISignalMappings.

[TPS_SYST_01120] Precedence of ISignalMappings ⌈ If a dedicated ISignalMapping for at least one ISignal within an ISignalGroup exists the mapping on the basis of shortNames is no longer applicable for any ISignal within that ISignalGroup. ⌋

[TPS_SYST_01121] Support of Mulicast signal routing ⌈ The 1:n multicast routing is supported with the definition of several ISignalMappings. See also the COM Signal Gateway fan-out description in section 6.10.1.2. ⌋

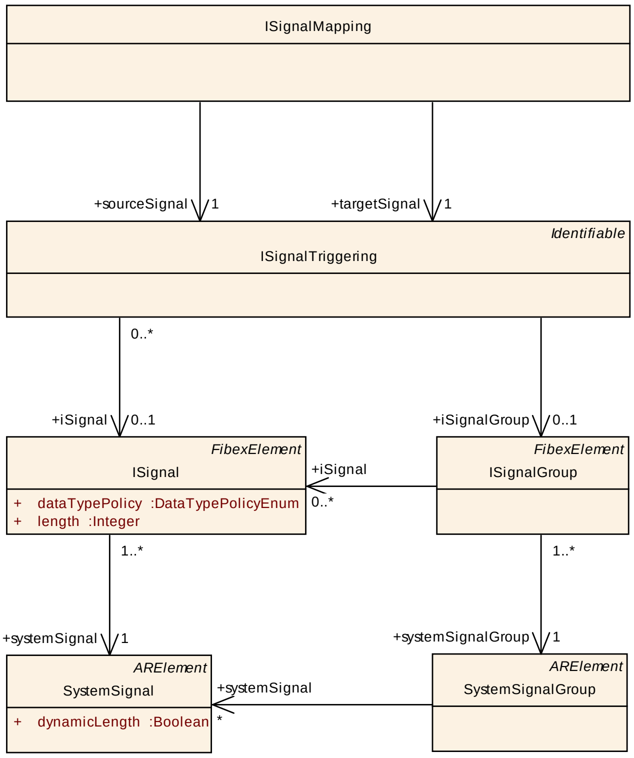

Figure 7.4: Signal Mapping (Fibex4Multiplatform: Signal Mapping)

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	***Datatype***	***Mul.***	***Kind***	***Note***
introduction	Documentation Block	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 7.7: ISignalMapping

7.3.1 Partial Signal Group Mapping

[TPS_SYST_01122] partial routing between `ISignalGroup`s ⌈ The `ISignalMapping` supports partial routing between `ISignalGroup`s which have not identical set of `ISignal`s within an `ISignalGroup`. ⌋

[constr_3053] Complete `ISignalMapping` of target `ISignalGroup` ⌈ If an `ISignalGroup` is referenced by a `targetSignal` there shall exist either an explicit or an implicit mapping (see [TPS_SYST_01120] for each contained ISignal of that ISignalGroup. ⌋

Figure 7.5 shows an example for a partial signal group mapping with explicit mappings for the GroupSignals.

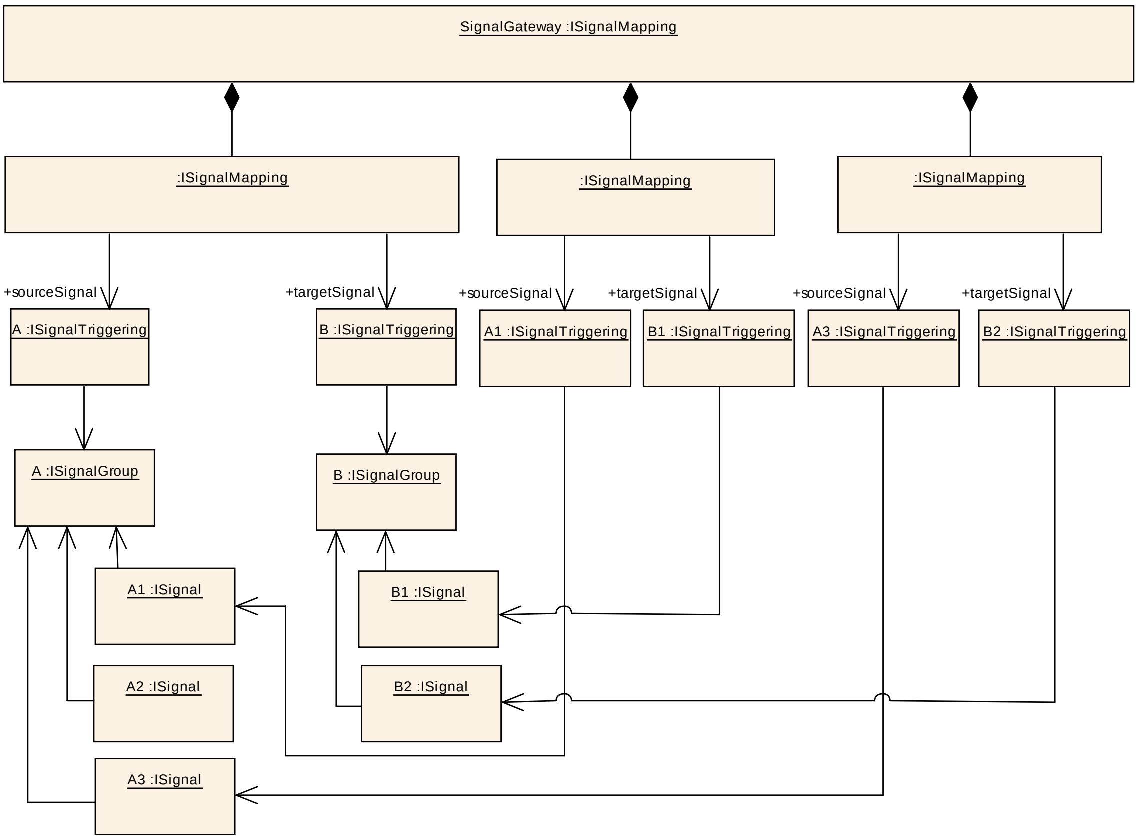

Figure 7.5: Partial Signal Group Mapping Example

8 Usage of the System Template

As introduced in [TPS_SYST_01003] the System Template is used to describe a System with category SYSTEM_CONSTRAINT_DESCRIPTION, a System with category ABSTRACT_SYSTEM_DESCRIPTION and a System with category SYSTEM_DESCRIPTION. System with category SYSTEM_EXTRACT is described in more detail in chapter 9. System with category ECU_EXTRACT is described in more detail in chapter 10.

Certain elements of the System Template may have a different meaning at the different stages of the AUTOSAR Methodology. The following sections describe the differences.

8.1 System Constraint Description

Meta-classes, Chapters	*Usage to describe the System Contraints*	*Usage to describe the System Configuration*
CommunicationCluster, EcuInstance (chapter 3)	The Topology is completely described in the System Constraint Description.	The Topology description will be unchanged copied to the System Configuration description. The Topology may only be changed during another iteration development step of the whole system.

Meta-classes, Chapters (cont.)	*Usage to describe the System Contraints (cont.)*	*Usage to describe the System Configuration (cont.)*
FrameTriggering, PduTriggering, ISignalTriggering (chapter 6)	The System with category SYSTEM_CONSTRAINT_DESCRIPTION describes all FrameTriggerings that are predefined on all CommunicationClusters of a vehicle. The predefinition of the communication matrix forces the system generator to use the given FrameTriggerings. Constraints for the system generator arise here e.g. from the used bus bandwidth, used identifiers as well as from the timing and at which position in a Frame a Pdu is transmitted on a PhysicalChannel on a CommunicationCluster. Such a manual definition of the communication can be made for any reason where it is necessary to restrict the system generator. One example is the usage of legacy EcuInstances in an AUTOSAR System. The FrameTriggerings that are transmitted or received by these legacy EcuInstances are constraints for the system generator because they cannot be changed, if the compatibility is supposed to be achieved without any changes at the legacy EcuInstances.	In contrary to the System with category SYSTEM_CONSTRAINT_DESCRIPTION the final System with category SYSTEM_DESCRIPTION contains all FrameTriggerings, PduTriggerings, ISignalTriggerings that will be sent by any EcuInstance in the car. No matter if they were predefined (system constraint) or if they were generated by the system generator. The available information, in addition to the information, which is inserted by the AUTOSAR Ecu configuration generator step, will be used as input to configure the Basic SW for the communication.

Meta-classes, Chapters (cont.)	*Usage to describe the System Contraints (cont.)*	*Usage to describe the System Configuration (cont.)*
Gateway (chapter 7)	The System with category SYSTEM_CONSTRAINT_DESCRIPTION describes all Gateways in the system including their IPduMappings and ISignalMappings that are predefined. The reasons for such predefinitions are quite the same as for the predefinitions of the FrameTriggerings.	In contrary to the System with category SYSTEM_CONSTRAINT_DESCRIPTION the final System with category SYSTEM_DESCRIPTION describes all Gateways with all their IPduMappings and ISignalMappings. No matter if they were predefined (System Constraint) or if they were generated by the System Generator.
SwcToEcuMapping (chapter 5.1.1)	The mapping of Software Components to EcuInstances may be predefined. The predefinition will force the system generator to use the specified mapping. Thus, with the SwcToEcuMapping element it is possible to describe that one or more Software Components must be mapped to a specific EcuInstance.	In a complete System with category SYSTEM_DESCRIPTION, all Software Components are mapped to EcuInstances.
MappingConstraint (chapter 5.1.3) ComponentClustering (chapter 5.1.3.1) ComponentSeparation (chapter 5.1.3.2)	There may be system constraints that limit the system generators freedom to map Software Components to arbitrary EcuInstances. These system constraints can be necessary e.g. for optimization and safety reasons to make additional guidelines for the System Generator.	After the mapping has been completed, the System with category SYSTEM_DESCRIPTION will contain mapping descriptions for all elements, and the mapping constraints are obsolete. But that does not mean that mapping constraints have to be deleted after the system generation step. By deleting the mapping constraints you would lose the information why a mapping of a Software Component to an EcuInstance is chosen.

Meta-classes, Chapters (cont.)	*Usage to describe the System Contraints (cont.)*	*Usage to describe the System Configuration (cont.)*
DataMapping (chapter 5.2) SenderReceiverToSignalMapping (chapter 5.2.1.1) SenderReceiverToSignalGroupMapping (chapter 5.2.1.2) ClientServerToSignalGroupMapping (chapter 5.2.1.4)	The System with category SYSTEM_CONSTRAINT_DESCRIPTION may describe the predefined mapping of Software Components to certain EcuInstances (see chapter 5.1.1). Only if such a mapping exists, it is reasonable to define the DataMapping of the data exchanged between the Software Components.	In contrary to the System with category SYSTEM_CONSTRAINT_DESCRIPTION the final System with category SYSTEM_DESCRIPTION shall contain all DataMapping definitions. No matter if they were predefined (system constraint) or if they were generated by the System-Generator.

Meta-classes, Chapters (cont.)	*Usage to describe the System Contraints (cont.)*	*Usage to describe the System Configuration (cont.)*
SignalPathConstraint (chapter 5.2.2) CommonSignalPath (chapter 5.2.2.1) ForbiddenSignalPath (chapter 5.2.2.2) PermissibleSignalPath (chapter 5.2.2.3) SeparateSignalPath (5.2.2.4)	It can be necessary e.g. for optimization and safety reasons to make additional guidelines for the System Generator, which specific way a VariableDataPrototype or ClientServerOperation should take in the network without defining in which Pdu and Frame it is transmitted.	SignalPathConstraints are not an obligatory part of the System with category SYSTEM_DESCRIPTION. In the final System with category SYSTEM_DESCRIPTION every ISignal is assigned to a Pdu and every Pdu is assigned to a Frame. Thereby the paths of VariableDataPrototypes or ClientServerOperations on the network are implicitly described. But that does not mean that the SignalPathConstraints have to be deleted after the system generation step. By deleting the SignalPathConstraints you would lose the information why you have chosen e.g. a specific mapping of an ISignal into a Pdu. If you extend or change the system at a later stage the missing SignalPathConstraints could lead to not wanted signal mappings by the System Generator.

Table 8.1: Usage of the System Template

8.2 Abstract System Description

[TPS_SYST_01134] Abstract System Description ⌈ Due to the fact that the functional view on vehicle system can differ from the actual technical definition of the software-architectures of individual EcuInstances the System Template optionally allows to define a System with category ABSTRACT_SYSTEM_DESCRIPTION. ⌋

[TPS_SYST_01135] Refactoring of an Abstract System Description into a project specific technical view of the software architecture ⌈ The System with category ABSTRACT_SYSTEM_DESCRIPTION concentrates on the functional aspects of the system design and provides an own abstract VFB. During the further activities this abstract view shall be refactored into a more project specific technical view of the software architecture.

It is important to note that during the refactoring of the `System` with `category` `ABSTRACT_SYSTEM_DESCRIPTION` into the `System` with `category` `SYSTEM_DESCRIPTION` no restrictions to the allowed actions apply (This is in contrast to the activity of deriving the `System` with `category` `SYSTEM_EXTRACT` from the `System` with `category` `SYSTEM_DESCRIPTION`, see section 9). ⌋

[TPS_SYST_01136] `ViewMapSet` and `ViewMap` are used to trace the transformations between different models ⌈ The `ViewMapSet` and `ViewMap` elements are used to trace the transformations between different models within the AUTOSAR environment. ⌋

These classes are described in more detail in the Generic Structure Template [2].

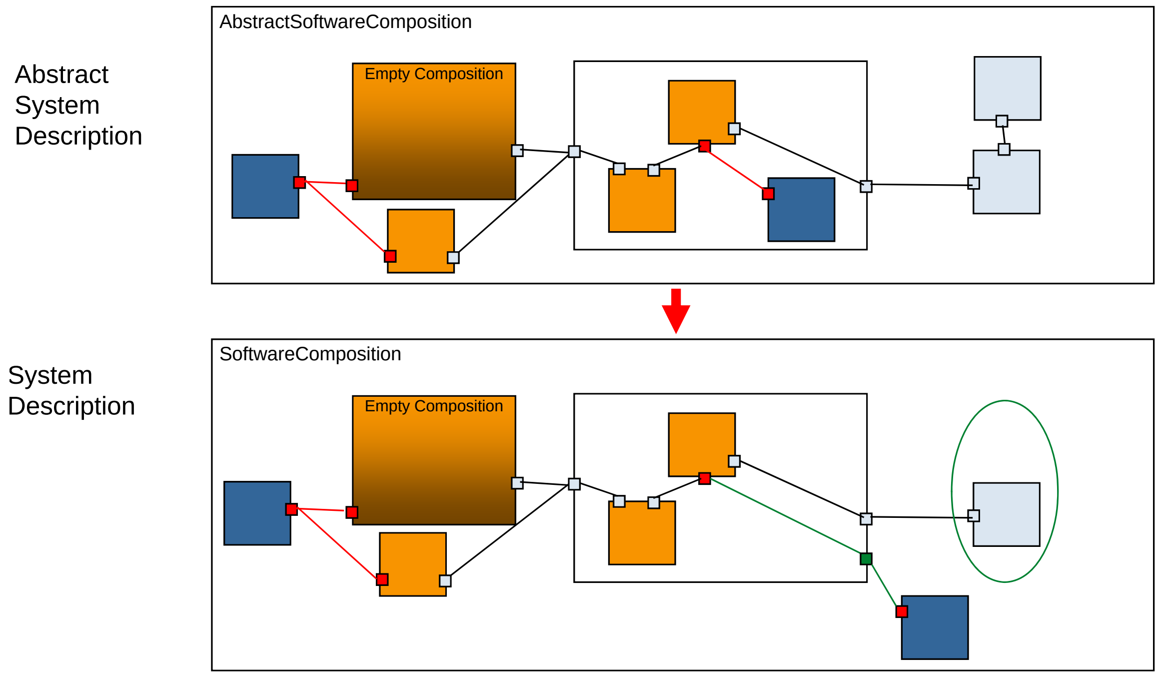

Figure 8.1: Abstract System Description refactoring to a System Description

9 System Extract of the System Configuration Description

This chapter describes contents and creation of the AUTOSAR work product System with category SYSTEM_EXTRACT, based on Meta Model elements contained in the System Template and Software Component Template.

The System with category SYSTEM_EXTRACT is introduced to allow a collaboration between an OEM and a Supplier.[1] The OEM/Supplier Collaboration scenario is described in more detail in chapter 9.1.

The OEM is often only interested in the required functionality and the integration of the functionality into the System. Thus the OEM provides a basis for designing a subsystem, which is developed by the supplier. One difference to the System with category ECU_EXTRACT is that the System with category SYSTEM_EXTRACT is not fully decomposed and still needs to be refined before it forms the basis for the ECU configuration. Another difference is that a System with category SYSTEM_EXTRACT is not fixed to an EcuInstance.

[TPS_SYST_01123] System Extract may cover one or many EcuInstances ⌈ The System with category SYSTEM_EXTRACT may cover one or many EcuInstances.
⌋

The System with category SYSTEM_EXTRACT is using the same meta model elements as the System with category SYSTEM_DESCRIPTION. The System with category SYSTEM_DESCRIPTION is a special case of a System with category SYSTEM_EXTRACT. From the technical point of view there is no difference. The distinction is only made for the sake of Methodology [4].

In the System with category SYSTEM_EXTRACT the OEM strips all information from the System with category SYSTEM_DESCRIPTION that is not needed for the definition of the subsystem. There is one exception to this simple ”remove” rule: the communication mapping may need to be extended, which will be described in more detail in chapter 9.2.

[TPS_SYST_03000] Co-existing System with category SYSTEM_DESCRIPTION and System with category SYSTEM_EXTRACT ⌈ In order to be able to handle one System with category SYSTEM_DESCRIPTION and one or several Systems with category SYSTEM_EXTRACT within the same workspace it shall be possible to provide different full qualified names to the elements of System with category SYSTEM_EXTRACT. ⌋*(RS_SYST_00045)*

When different Systems with various categories co-exist it is possible to define ViewMap and ViewMapSet between their elements according to [TPS_SYST_01136].

[1] Collaboration scenarios between different departments of an OEM are also supported by the System with category SYSTEM_EXTRACT. For the sake of simplicity such scenarios are not addressed here.

In contrast to the System with category ECU_EXTRACT the System with category SYSTEM_EXTRACT may contain CompositionSwComponentTypes. Empty CompositionSwComponentTypes in the System with category SYSTEM_EXTRACT represent subsystems that need to be refined by a Supplier. Figure 9.1 shows an example where a System with category SYSTEM_DESCRIPTION is stripped down to a subsystem.

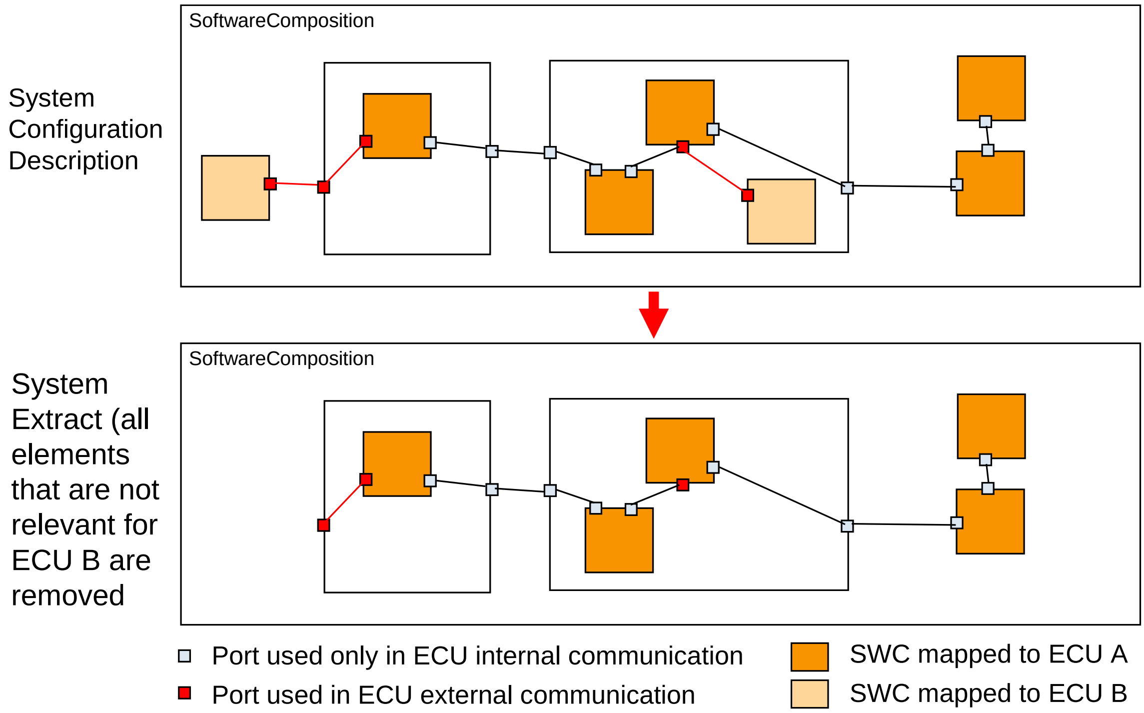

Figure 9.1: System Extract creation: irrelevant elements are removed from the System Description

9.1 OEM/Supplier Collaboration Scenario

In an important collaboration scenario, an OEM commissions a supplier to provide implementations of one or more functionalities to be integrated into an AUTOSAR system in the form of Application Components. The OEM is primarily interested in the required functionality and the interfaces defining the integration of the Software Component into the System VFB rather than the internal structure of such a component. On the other hand, the supplier, delivering both the component implementation in combination with the ECU it is destined to run on, may claim the internal structure of such a higher-level component contains substantial intellectual property, and hence may not want to disclose its internal works to the OEM.

Effectively, the use case can be described in the following manner:

- The OEM generates a System with category SYSTEM_EXTRACT from the System with category SYSTEM_DESCRIPTION. From the System with category

SYSTEM_DESCRIPTION all elements are removed that are not relevant for the design of the subsystem, such as SW components or topology elements.

- The OEM can deliver a sub-structure of Software Compositions or even Atomic Software Components in the System with category SYSTEM_EXTRACT. But the System with category SYSTEM_EXTRACT can also contain empty Software Compositions. The OEM shall have the possibility to define only the outer shell of a Software Composition that is to implement a certain functionality. Such an empty CompositionSwComponentType does contain all the provided and required ports with the included ReceiverComSpecs and SenderComSpecs describing the requested component's outside communication needs. But it does not need to contain SwComponentPrototypes or SwConnectors at this stage.
- Such empty components are added to a System's VFB, the outside ports are connected with other components in the VFB. However, at this stage the inner structure of such CompositionSwComponentType can still be left empty.
- The System with category SYSTEM_EXTRACT contains the mapping of components to the target EcuInstances, including the empty compositions. Signal mappings affecting the empty compositions are targeting the CompositionSwComponentType's ports.
- The OEM delivers the System with category SYSTEM_EXTRACT to the Supplier.
- The Supplier adds the substructure to the empty CompositionSwComponentTypes by adding SwComponentPrototypes and SwConnectors. This once more leads to a hierarchical VFB, effectively the Supplier creates a local System Description for his subsystem.
- The Supplier adjusts the Signal mappings to the actual ports of the inner AtomicSwComponentType prototype.
- The Supplier generates the System with category ECU_EXTRACT from his ECU-local system description. The resulting System with category ECU_EXTRACT does not include prototypes of type CompositionSwComponentType any longer.
- Based on this System with category ECU_EXTRACT the actual ECU configuration is done.

When the supplier receives the System with category SYSTEM_EXTRACT from the OEM he has basically two choices how to proceed:

1. The Supplier takes the System with category SYSTEM_EXTRACT of the OEM as the structural basis for the ECU development. In this case the following steps may follow:
    - The Supplier adds the substructure to the empty CompositionSwComponentTypes by adding SwComponentPrototypes and SwConnectors. This once more leads to a hierarchical VFB, effectively the Supplier cre-

ates a local System Description for his subsystem (System with category ECU_SYSTEM_DESCRIPTION).

- The Supplier adjusts the Signal mappings to the actual ports of the inner AtomicSwComponentType prototype.

2. The Supplier creates an own structure to base the ECU development on System with category ECU_SYSTEM_DESCRIPTION and perform a view mapping between the OEM's System with category SYSTEM_EXTRACT and the System with category ECU_SYSTEM_DESCRIPTION. In this case the following steps may follow:

   - The Supplier develops an own structure how the ECU shall be designed but needs to respect the required outer boundary of the OEM's required communication behavior (ReceiverComSpecs and SenderComSpecs).

   - The Supplier adjusts the Signal mappings to the actual ports of the inner AtomicSwComponentType prototype.

When the design of the System with category ECU_SYSTEM_DESCRIPTION is complete the following steps follow:

- The Supplier generates the System with category ECU_EXTRACT from his System with category ECU_SYSTEM_DESCRIPTION. The resulting System with category ECU_EXTRACT does not include prototypes of type CompositionSwComponentType any longer.

- Based on this System with category ECU_EXTRACT the actual ECU configuration is done.

9.2 Data Mapping in the System Extract

As mentioned before, there is a slight complication to the simple "remove" rule. This can be shown best with an example.

Example: Assume a simple topology with two EcuInstances A and B and three Pdus X (sent from A to B), Y (sent from B to A) and Z (sent from B to A) as shown in Figure 9.2.

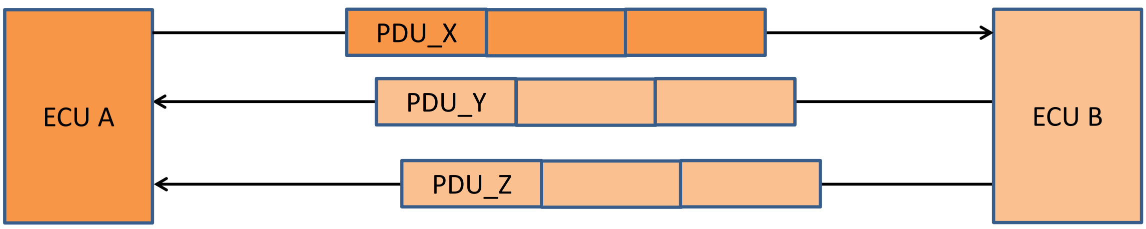

Figure 9.2: Example topology with two EcuInstances and three Pdus exchanged between them

Furthermore assume a composition of software-components realized by the meta-class CompositionSwComponentType as shown in Figure 9.3. It consists of six SwComponentPrototypes 'A1' to 'A3' (aggregated in composition 'SwCompA'), 'B1' / 'B2' (aggregated in composition 'SWCompB'), 'C1' (aggregated in composition 'SWCompAplusBplusC') and an empty composition 'SWCompC'.

The overall composition 'SWCompAplusBplusC' aggregates 'SwCompA', 'SWCompB', the empty 'SWCompC' and the SwComponentPrototype 'C1'.

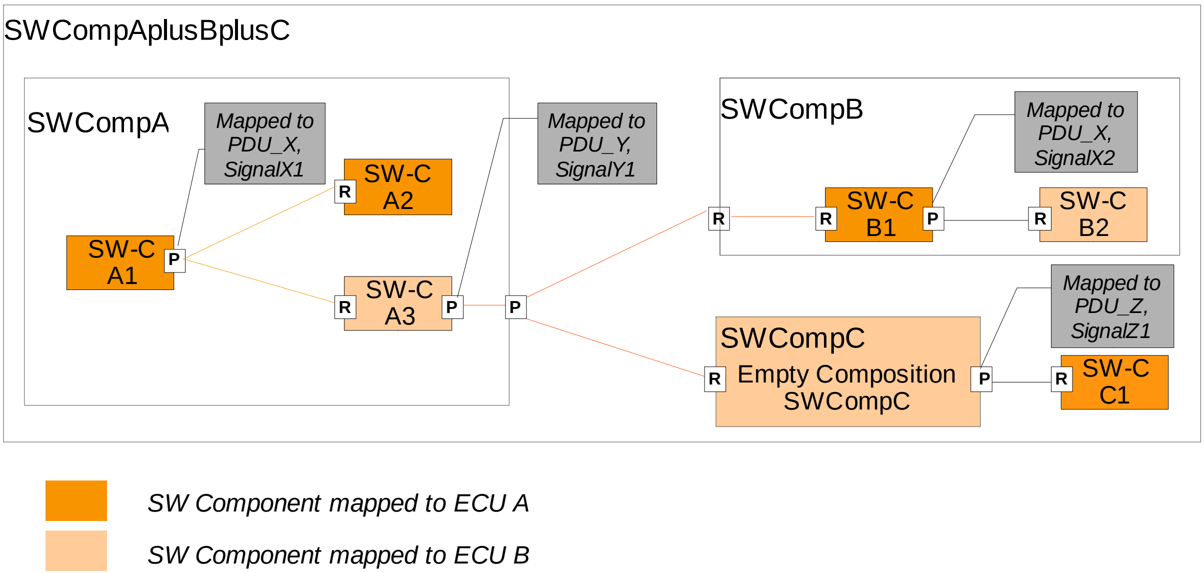

Figure 9.3: Example SW composition with mapping information

The atomic SwComponentPrototypes 'A1', 'A2', 'B1' and 'C1' are mapped to 'ECU A'. The atomic SwComponentPrototypes 'A3', 'B2' and the empty composition 'SWCompC' are mapped to 'ECU B'. The data sent from

- 'A1' to 'A3' is mapped to 'PDU_X', 'SignalX1',
- 'B1' to 'B2' is mapped to 'PDU_X', 'SignalX2' and
- 'A3' to 'B1' and 'A3' to 'SWCompC' is mapped to 'PDU_Y', 'SignalY1'
- 'SWCompC' to 'C1' is mapped to 'PDU_Z', 'SignalZ1'

As usual, the data mapping rules refer to the VariableDataPrototype in the PPortPrototype of the sending SW component. Note that DataMappings can be performed on compositions and on atomic SwComponentPrototypes as described in chapter 5.2.1. [2]

Figure 9.4 shows how the System extract for ECU A and for ECU B of this SW composition would look like: Only those elements are included that are relevant for the subsystem.

[2]Data mapping is allowed on empty compositions and on compositions that contain atomic SwComponentPrototypes.

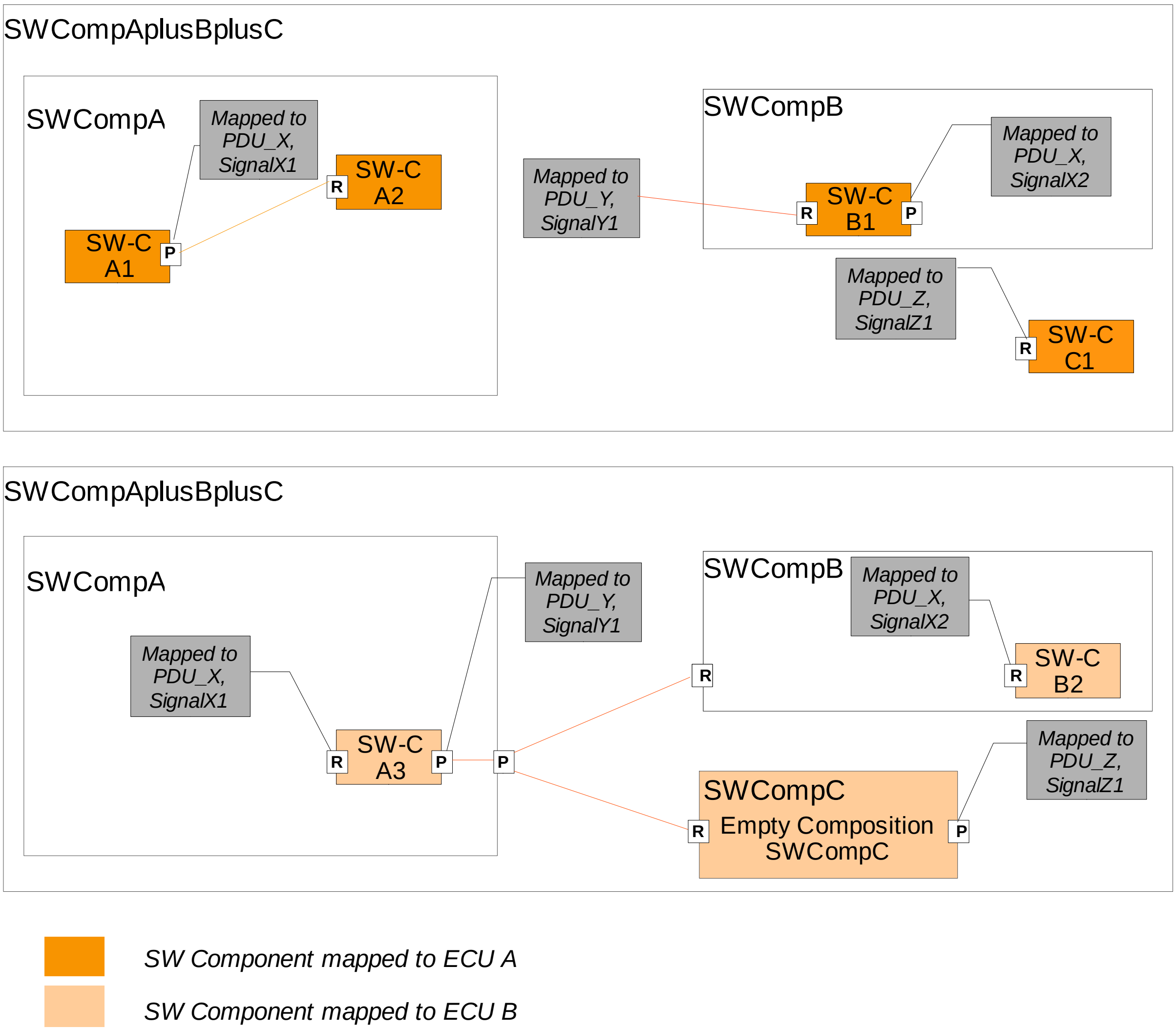

Figure 9.4: Example System extract for ECU A (upper figure) and ECU B (lower figure) of above introduced composition

In both figures all SwComponentPrototypes and compositions that are mapped onto the EcuInstance are included. The SwConnector between these SwComponentPrototypes are also included. Furthermore, the relevant topology information and communication matrix have to be included, but they are out of scope of this example.

SwConnectors that were used to connect to SW components that are not included in the System Extract are not included. Instead, the mapping to an ISignal in a Pdu is used to identify the source/destination of that data.

The problem that new mapping rules have to be added arises for example in the System Extract for 'ECU A' with the mapping to 'PDU Y', 'SignalY1': Since SW component 'A3', which was referenced in the original mapping, is no longer included, the data mapping needs a new data element in a port to reference to. In the example, it is the required port of 'B1', so that the Supplier has the information that B1 receives the data via 'PDU Y'.

9.3 SW component inclusion and top level data mapping

In section 9.2 the approach is to provide the DataMapping on the PortPrototypes of the SwComponentPrototypes which are mapped to one EcuInstance. Since the granularity of mapping SwComponentPrototypes to EcuInstances is possible for individual atomic SwComponentPrototypes this approach may result in many DataMappings from different software component PortPrototypes to the same SystemSignal (depending where in the hierarchical structure they are located).

An alternative approach is to provide the complete communication information of the whole System Extract on the RootSwCompositionPrototype and perform the DataMapping on the PortPrototypes of the RootSwCompositionPrototype only. This approach is illustrated in figure 9.5.

PortPrototypes are created on the RootSwCompositionPrototype representing the external communication of this EcuInstance. DelegationSwConnectors are created to establish the communication of the external software components with the software components inside the local EcuInstance.

In figure 9.5 the software components X, Y and Z are mapped to remote EcuInstances. Their communication needs are collected in PortPrototypes on the RootSwCompositionPrototype and the communication is delegated via SwConnectors inside the hierarchical software component structure.

In this example the approach for X and Y is trivial since there are only some DelegationSwConnectors required to connect the PortPrototypes of the RootSwCompositionPrototype with the PortPrototypes of the respective SwComponentPrototypes.

But for SwComponentPrototype Z the approach needs to be extended, because the communication on system level is designed to happen inside the composition V. In this case the communication needs to be delegated out of the composition (creation of DelegationSwConnectors inside the composition V) to be visible in the RootSwCompositionPrototype. Then again the approach of connection to the RootSwCompositionPrototype can be applied.

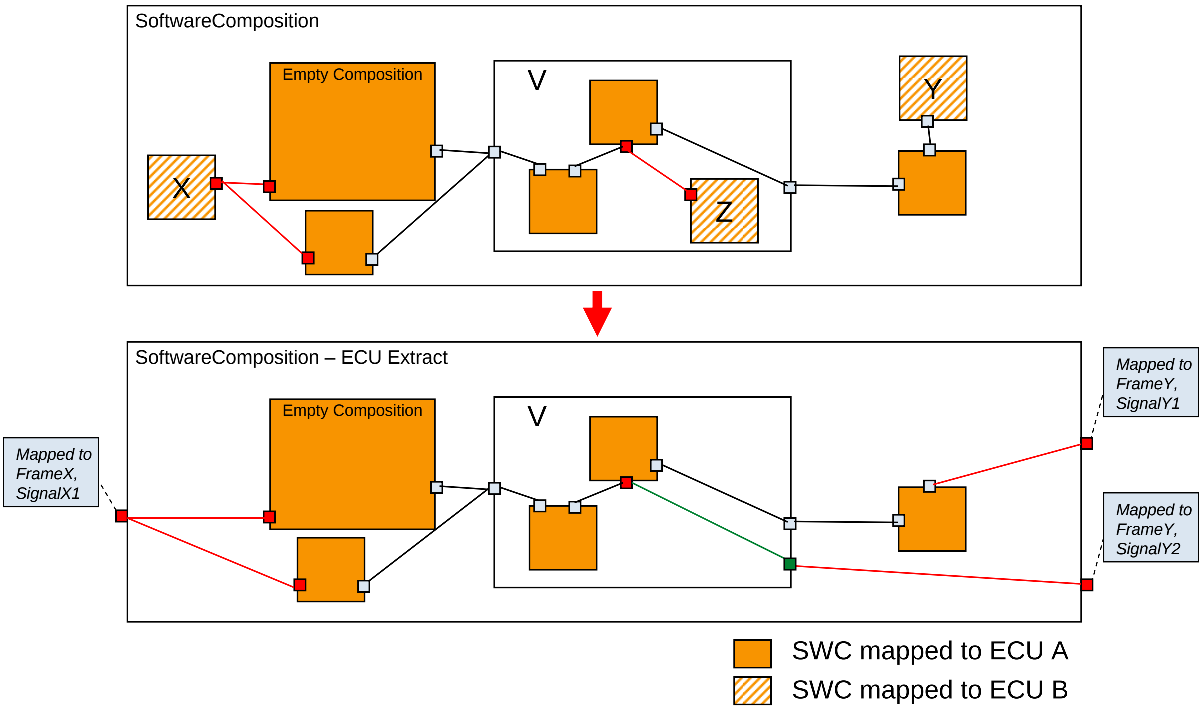

Figure 9.5: Example with software components mapped to two ECUs

10 ECU Extract of the System Configuration Description

This chapter describes contents and creation of the AUTOSAR System with category ECU_EXTRACT, based on Meta Model elements contained in the System Template and Software Component Template.

The System with category ECU_EXTRACT represents the view of one specific EcuInstance onto the overall System with category SYSTEM_DESCRIPTION. The System with category ECU_EXTRACT forms the basis for configuring that particular EcuInstance in focus.

For instance, RTE configuration fundamentally depends on the number and types of SwComponentPrototypes deployed onto the EcuInstance; Services are configured according to those Software Components' ServiceNeeds; the COM-stack BSW modules will be configured considering the EcuInstance's participation in the overall System Network Topology and Communication.

[TPS_SYST_01139] Ecu Extract covers exactly one EcuInstance ⌈ The System with category ECU_EXTRACT shall only contain the subset of information derived from the System with category SYSTEM_DESCRIPTION or System with category SYSTEM_EXTRACT relevant for configuring the targeted EcuInstance. ⌋

In order to keep ECU configuration focused and manageable despite the complexity of a full System Configuration, all other information shall be stripped from the System with category SYSTEM_DESCRIPTION or from the System with category SYSTEM_EXTRACT when creating the System with category ECU_EXTRACT.

AUTOSAR VFB Descriptions naturally form hierarchies of CompositionSwComponentTypes. Consequently, in the System Configuration the SWC-related information for different EcuInstances is not separated but in general is intermingled. In contrast, for the task of ECU configuration (RTE configuration, Service Configuration, Measurement and Calibration) a hierarchically "flat view" on the SwComponentPrototypes running on the EcuInstances is preferable over a hierarchical view, which is more favored by application-software development. Thus, deriving an System with category ECU_EXTRACT actually is a model transformation, following a set of rules described in the following sections.

As System- and ECU development typically happens in iterations, the use case of repeatedly extracting the information from an incrementally changing System Configuration needs to be considered. In particular, it must be possible to detect changes between consecutively generated ECU extracts in order to selectively update the existing ECU configuration (10.5).

AUTOSAR supports the definition and consequently the handling of Variability in the System Configuration. According to the specified binding time associated with a particular VariationPoint, typically some of these variants will already be resolved at the time of a System with category ECU_EXTRACT. If however the binding time occurs in a later stage of the AUTOSAR methodology, i.e. during ECU Configuration or later, the variability needs to be carried over to the System with category ECU_EXTRACT. This also holds true for Variation points that ultimately are resolved at system configuration time but affect post-build configuration parameters. (10.6)

The System with category ECU_EXTRACT logically forms one entity. Therefore, for ease of readability the rest of the chapter assumes just one file, “the XML file”. However, it explicitly is allowed to split the System with category ECU_EXTRACT over several files.

10.1 Topology

Only those Topology elements relevant for the EcuInstance in scope are taken over from the System with category SYSTEM_DESCRIPTION into the System with category ECU_EXTRACT.

- The System with category ECU_EXTRACT is always associated with exactly one EcuInstance. Therefore exactly one EcuInstance is included along with all classes included in EcuInstance by composition: CommunicationControllers and CommunicationConnectors with all their CommConnectorPorts.
- A CommunicationCluster is included along with all its PhysicalChannels if at least one PhysicalChannel is used by the EcuInstance. In other words, if at least one of the included CommunicationConnectors is referenced by any of a CommunicationCluster's PhysicalChannels, the whole CommunicationCluster and all its PhysicalChannels are included.
- From the used PhysicalChannels, only those FrameTriggerings, PduTriggerings, ISignalTriggerings shall be included that are used by the EcuInstance, e.g. they are associated with a FramePort, IPduPort, ISignalPort belonging to one of the EcuInstance's CommunicationConnectors. *Note:* Including just a subset of a PhysicalChannel's FrameTriggerings, PduTriggerings, ISignalTriggerings is possible without changing the PhysicalChannel itself because of the ≪splitable≫ stereotype applied on the PhysicalChannel / FrameTriggering, PduTriggering, ISignalTriggering composition.

As the Topology elements are not modified when taken over into the System with category ECU_EXTRACT, their package structure and short names are not touched (see section 10.4.1).

10.2 Top-level Software Composition

In the System with category SYSTEM_DESCRIPTION the application software composition is hierarchic by nature as described in chapter 4. When mapping SwComponentPrototypes onto concrete EcuInstances using the SwcToEcuMapping class (section 5.1.1), either SwComponentPrototypes of type AtomicSwComponentType, or SwComponentPrototypes of type CompositionSwComponentType are deployed onto one specified EcuInstance.

In order to obtain this ECU-centric view, the hierarchical structure of the System with category SYSTEM_DESCRIPTION needs to be transformed into a 1-layer representation, where one distinguished CompositionSwComponentType hosts all SwComponentPrototypes of type AtomicSwComponentType to run on the EcuInstance. In the System with category ECU_EXTRACT the resulting RootSwCompositionPrototype is a flat structure where the included SwComponentPrototypes become real SWC instances, reflecting the actual resource needs on the targeted EcuInstance.

[TPS_SYST_01140] Ecu Extract contains only SwComponentPrototypes of type AtomicSwComponentType in the RootSwCompositionPrototype ⌈ The System with category ECU_EXTRACT only contains SwComponentPrototypes of type AtomicSwComponentType in the RootSwCompositionPrototype which are effectively mapped onto the EcuInstance in focus. ⌋

The transformation from hierarchical to flat Software Component structure includes a number of steps, to be performed per ECU. The list below outlining this process assumes that the extraction is done for the first time; if an System with category ECU_EXTRACT already exists from a previous development cycle, the extract shall merely be updated instead of created; for more details on iterative development see section 10.5.

- Create the one CompositionSwComponentType which will represent the ECU's SW subsystem (in further steps referred to as ECU flat view)
- To this ECU flat view, add a SwComponentPrototype for each instance of any AtomicSwComponentType mapped onto the EcuInstance. Copy all the identifiable information from the originating SwComponentPrototype, but assign an unique short name to the new element. The newly created SwComponentPrototypes are typed by the original AtomicSwComponentType.
- Unroll the connector paths leading to and from the included components:
  - For ECU internal communication, use AssemblySwConnector to connect PortPrototypes.
  - For ECU external communication, add delegated PortPrototypes to the ECU flat view CompositionSwComponentType. Each delegated PortPrototype shall be connected via a DelegationSwConnector with PortPrototypes of the included SwComponentPrototypes that are used for the external communication.
  - VariableDataPrototypes and ClientServerOperations of the delegated PortPrototypes are mapped to SystemSignals.

- If the System with category SYSTEM_DESCRIPTION prescribes an Implementation for a SwComponentPrototype by using SwcToImplMapping, a corresponding constraint needs to be created in the System with category ECU_EXTRACT of the targeted EcuInstance. The SwcToImplMapping's component reference needs to be adjusted to the flat representation, while maintaining the original reference to the Implementation.
- ComSpecs on the PortPrototypes of the FlatView composition shall be taken over from the PortPrototypes of the top-most composition during flattening. Rationale: This is an approach to avoid the existence of multiple ComSpecs for a dataElement/clientServerOperation. Please note that this approach is a short term solution for specific use-cases. A harmonized solution for the usage of ComSpecs will be provided in future.

Figure 10.1 illustrates the process of flattening the hierarchical Software Composition into an ECU Flat View representation, as outlined in the previous paragraphs. The following sections explain the concrete transformation steps in more detail.

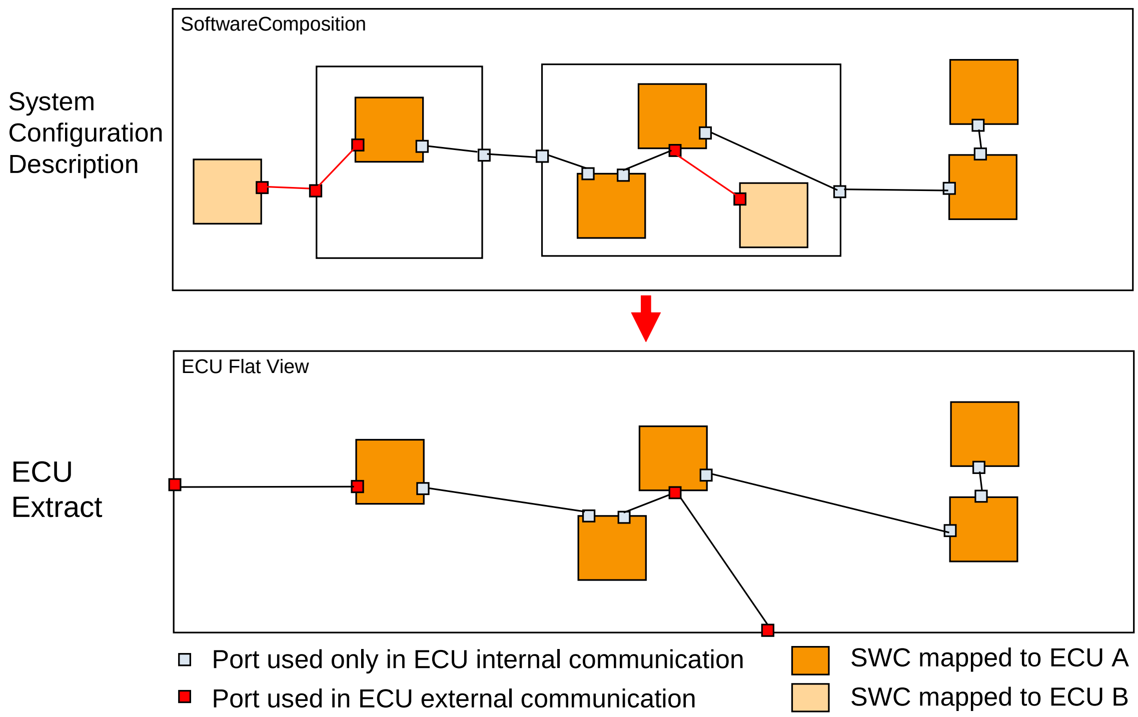

Figure 10.1: Flattening of a hierarchic Software Composition into an ECU Flat View, and the distinction between ports used in internal and those used in external communication.

Please note that instantiation specific scheduling of runnables shall be maintained when generating a System with category ECU_EXTRACT. This maintenance covers the rewrite of the instanceRef to the RTEEvent respectively the aggregation of the instantiationRTEEventProps to the next CompositionSwComponentType.

10.2.1 ECU Flat view

The first step of extracting the ECU specific Software View is the creation of a new CompositionSwComponentType (further referred to as ECU flat view). This new element serves as a container for collecting all SwComponentPrototypes of type AtomicSwComponentType deployed on the EcuInstance. In order to include the ECU flat view into the actual System with category ECU_EXTRACT, the System must have its child class RootSwCompositionPrototype pointing to this ECU flat view.

Next, all SwcToEcuMappings present in the System with category SYSTEM_DESCRIPTION need to be analyzed according to the precedence rules (Section 5.1.1) in order to establish the exact set of AtomicSwComponentType instances to be included on this EcuInstance.

For each of these component instances, regardless of their order of depth in the System Configuration Description's Component hierarchy, exactly one SwComponentPrototype shall be created in the ECU flat view CompositionSwComponentType. The new element's description and type information shall be taken over from the original SwComponentPrototype as present in the System with category SYSTEM_DESCRIPTION. As an important exception to this rule, the SwComponentPrototype's shortName must be unique in the name space formed by the ECU flat view.

The special case of prototypes of ParameterSwComponentTypes and ServiceProxySwComponentTypes is treated in almost the same way. The only difference is that these component types can be instantiated at most once per EcuInstance and that for a given prototype in the System, instances on several EcuInstances can be created. The replication of ParameterSwComponentTypes and ServiceProxySwComponentTypes on several EcuInstances does not require any special treatment of their communication properties. For ParameterSwComponentTypes there are SwConnectors defined but no communication is involved. For more details see [TPS_SWCT_01422] in the SwComponentTemplate [5].

10.2.2 Internal Communication

When flattening the RootSwCompositionPrototype for the System with category ECU_EXTRACT, not only all of the ECU's Software Components are to be collected in the ECU flat view, but also any connection existing between PortPrototypes of the included SwComponentPrototypes needs to be projected onto the same RootSwCompositionPrototype.

In the hierarchical RootSwCompositionPrototype, communication between Software Components is specified by a combination of AssemblySwConnectors and DelegationSwConnectors. Several DelegationSwConnectors may be combined in case of a multiple-level delegation, however there will always be exactly one AssemblySwConnector on the outermost CompositionSwComponentType the port is delegated to.

In the ECU flat view, any such number of stringed together SwConnectors effectively connecting two PortPrototypes of SwComponentPrototypes mapped to the same EcuInstance are resolved to exactly one AssemblySwConnector per connected port pair. As there are no additional levels of "inner SwComponentPrototypes". DelegationSwConnectors are only used to display the outside communication of an ECU in the ECU flat view.

[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 VariableDataPrototypes specified by the SenderReceiverInterface of the RPortPrototype need to be supplied by some of the PPortPrototypes being connected with SwConnectors. ⌋

For the System with category SYSTEM_DESCRIPTION, the Software Component Template Specification [5] allows a CompositionSwComponentType's outer PortPrototype to be connected to more than one inner port, observing a set of compatibility rules between the outer and the inner port's SenderReceiverInterfaces. Such a "merge" and "split" functionality for mixing VariableDataPrototypes is used to limit the number of SwConnectors required to connect PortPrototypes on higher VFB levels and thus reduce complexity in the wiring of such higher-level CompositionSwComponentTypes. On the other hand this means that an AssemblySwConnector in a hierarchical VFB may expand to more than one Port-Port pair. Naturally, in the ECU flat view such "hidden" additional connections need to be made explicit by unrolling them into concrete AssemblySwConnectors.

Additionally PassThroughSwConnector may be used to map PortInterface elements between require and provide outer ports of CompositionSwComponentTypes in order to use RTE features for mapping or conversion instead of real software components. The following paragraph suggests a way how such an unrolling of SwConnectors may be accomplished.

Starting with the top-level RootSwCompositionPrototype indicating the outermost CompositionSwComponentType, the hierarchical software model of SwComponentPrototypes is recursively iterated; for each prototype of CompositionSwComponentType, all its AssemblySwConnectors are being iterated. For each such found AssemblySwConnector both connector ends are evaluated for DelegationSwConnectors further delegating the connection: In order to consider the use cases of signal "merge" and "split", all possible communication partners need to be identified, recursively following DelegationSwConnectors in both directions. For each identified pair of PPortPrototypes and RPortPrototypes *actually exchanging* Information one AssemblySwConnector will be created in the ECU flat view.

In case that a string of SwConnectors started by AssemblySwConnector connects - directly or via DelegationSwConnectors - to a PassThroughSwConnector the SwConnector string is conjunct with the SwConnector string of the other end of the PassThroughSwConnector. Please note that the "merge" and "split" capability of DelegationSwConnectors and PassThroughSwConnectors requires an individual treatment of the single PortInterface elements for the evaluation of the SwConnector string.

The following rules must be followed when PortInterfaceMappings are converted for the flat view. PortInterfaceMappings supports the connection of Ports typed by two different PortInterfaces with unequal named PortInterface elements. More details can be found in [5].

- When unrolling a string of SwConnectors into a single SwConnectors all compatibility rules and PortInterfaceMappings of the individual SwConnector need to be considered for determining which VariableDataPrototypes are being transferred between provider and requester. If VariableDataPrototypes are to be filtered out a PortInterfaceMapping shall be provided to the flatten connector such that only the transfered VariableDataPrototypes are included in the mapping.
- When unrolling a string of SwConnectors into a single SwConnector all of the PortInterfaceMappings of the individual SwConnectors need to be considered for combining them into a single PortInterfaceMapping to be associated with a new SwConnector.

10.2.3 External Communication

In a System with category SYSTEM_DESCRIPTION , whenever two SwComponentPrototypes are specified to communicate across EcuInstances, the details of this communication need to be fully specified: VariableDataPrototypes of SenderReceiverInterfaces and ClientServerOperations of ClientServerInterfaces are mapped onto SystemSignals as carriers of information transported across the network. According to 5.2, each instance of a AutosarDataPrototype that is to be sent over AUTOSAR COM shall be mapped exactly once onto its individual SystemSignal, regardless of how many components receive the information or over how many PhysicalChannels the SystemSignal is transported.

As described above, deriving the System with category ECU_EXTRACT from System with category SYSTEM_DESCRIPTION or from System with category SYSTEM_EXTRACT means that all SwComponentPrototypes to be included in the Ecu extract are recreated in an ECU flat view. Consequently, each DataMapping concerning a SwComponentPrototype to be mapped onto the EcuInstance requires that a corresponding DataMapping be created in the System with category ECU_EXTRACT.

The ECU flat view contains delegated PortPrototypes to display the outside communication of an EcuInstance. VariableDataPrototypes and ClientServerOperations of these delegated PortPrototypes are mapped to SystemSignals. The original instance references indicating the mapped AutosarDataPrototype need to be adjusted to the new “flat” location in the ECU flat view.

While for the System with category SYSTEM_DESCRIPTION it is sufficient to describe DataMappings only on the provider side, the System with category ECU_EXTRACT additionally requires such DataMappings on the requiring side's ports. In this case, a new DataMapping maps to the existing SystemSignal, previously defined in the System with category SYSTEM_DESCRIPTION on the provider side. This is explained in more detail in the following example, that is a continuation of the example from figure 9.3 in chapter 9.2:

[TPS_SYST_01145] PortInterfaceMappings in the ECU Extract ⌈ In the System with category ECU_EXTRACT the missing PortInterfaceMappings on the complementary side needs to be supplemented to DelegationSwConnectors. ⌋

Figure 10.2 shows how the System with category ECU_EXTRACT for ECU A of the SW composition that is defined in figure 9.3 would look like: Only those SwComponentPrototypes are included that are mapped to ECU A. The hierarchy present in the System with category SYSTEM_DESCRIPTION has been flattened into CompositionSwComponentType 'EcuAFlatView', including newly created SwComponentPrototype 'A1E', 'A2E', 'B1E' and 'C1E' for the component instances mapped to ECU A.

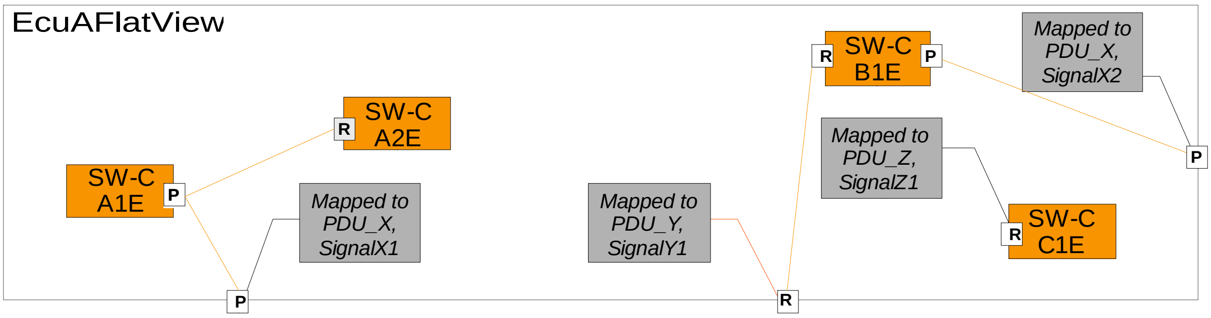

Figure 10.2: Example ECU extract for ECU A of above introduced composition

The SwConnectors to the outside ports (ECUFlatView composition ports) and SwConnectors that represent intra-ECU communication (in our example, only 'A1E' to 'A2E') are included. The VariableDataPrototypes and ClientServerOperations in the outside ports are mapped to SystemSignals. This DataMapping and the communication description is used to identify the source/destination of that data.

Furthermore, the relevant topology information and communication matrix have to be included, but they are out of scope of this example.

The problem that new mapping rules have to be added arises with the mapping to 'PDU_Y', 'SignalY1': Since SW component 'A3', which was referenced in the original mapping, is no longer included, the DataMapping needs a new VariableDataPrototype in a PortPrototype to reference to. In the example, the data of the required port of 'B1E' is referenced, so that the ECU generator has the information that 'B1E' receives the data via 'PDU_Y'.

10.2.4 Port Groups

A SwComponentType can optionally define PortGroups which allow to group PortPrototypes according to logical criteria, e.g. according to shared communication resources (see [5]). A PortGroup of a CompositionSwComponentType can be linked to "inner" PortGroups of the aggregated SwComponentPrototypes. Since the main purpose of this grouping is to configure the behavior of mode managers on an EcuInstance, this information must be preserved and broken down into the System with category ECU_EXTRACT.

The resulting CompositionSwComponentType in the ECU flat view will contain a set of PortGroups which refer to the linked inner port groups of the SwComponentPrototypes with AtomicSwComponentType. To get to this result, the following steps must be applied in the extraction process:

- Recursively ignore all PortGroups in CompositionSwComponentTypes in the hierarchical structure, which are not linked to any inner groups to be mapped on this EcuInstance.
- In the remaining structure of linked PortGroups find out the top level PortGroups (i.e. which are not referred by any higher level PortGroup on this EcuInstance) and put an element representing each top level PortGroup into the CompositionSwComponentType of the ECU flat view. This can result in name conflicts, which should be resolved by a suitable algorithm.
- Link these top level PortGroups to the inner PortGroups of the atomic component instances of the flat view according to the links found in the hierarchical structure. Naturally, the top level PortGroups in the ECU flat view are not directly referring any PortPrototypes and due to the first step they should be linked to at least one inner PortGroup.
- The PortGroups in SwComponentPrototypes with an AtomicSwComponentType on the EcuInstance should be unchanged.

10.2.5 Service Needs

Each software component might need services which are provided by the ECU Basic Software through AUTOSAR Services. ServiceNeeds are used to provide detailed information what the software component expects from the AUTOSAR Services when integrated on an actual ECU (see SWComponentTemplate [5] for more details). If an ECU Extract is created the following rules apply to the existing ServiceNeeds:

[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_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_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. ⌋

10.3 Communication

In explaining how SystemSignals are handled in the System with category ECU_EXTRACT, Section 10.2.3 touched on the topic of inter-ECU Communication. However, in order to enable the ECU Configuration of the COM-Stack, the relevant information of all layers of the AUTOSAR COM-Stack needs to be present in the System with category ECU_EXTRACT, including the central Communication classes ISignal, Pdu and Frame.

The above mentioned Communication elements have dependencies on each other, for ordinary COM-communication this means:

- Frames are assembled from one or more Pdus.
- ISignalIPdus carry their information in form of ISignals.
- ISignals as interaction points between RTE and COM refer to SystemSignals.

Note that the above list is not complete; TP and NM require additional elements. However, for the sake of clarity the following paragraphs describes the standard use case of a direct Signal-based communication between two EcuInstances. Once the handling of this case is understood, the additional model elements as NPdu, NmPdu, System-SignalGroup etc. can be handled following the same basic principles.

For the System with category ECU_EXTRACT only the ECU-relevant subset of information present in the system-wide communication is to be considered. In order to establish this set of information, the dependencies in the list above are being followed.

10.3.1 Frame

In a complete System with category SYSTEM_DESCRIPTION, every outside communication of an EcuInstance will either be associated with an outgoing or and incoming Frame. The exact number and types of Frames to be received or sent by an EcuInstance is determined by the Communication Matrix (Chapter 6).

According to the selection rules for the Topology (10.1), the System with category ECU_EXTRACT contains all FrameTriggerings associated with Frames that are of any interest to the EcuInstance: If a particular FrameTriggering refers to a FramePort of type 'out' the associated Frame is to be sent by the EcuInstance, if it refers to an 'in' port the Frame is to be received. Therefore, the following selection rule applies:

- The System with category ECU_EXTRACT shall contain all Frame elements which are referenced by any included FrameTriggering.

10.3.2 PDU

Frames are assembled from one or more Pdus. In order to include all required Pdu elements, the following selection criteria apply:

- The System with category ECU_EXTRACT shall contain all Pdu elements which are referenced by any included Frame's PduToFrameMapping.
- The System with category ECU_EXTRACT shall contain all Pdu elements which are referenced by any included PduTriggering.
- For multiplexed Pdus, additionally all ISignalIPdus referenced by the MultiplexedIPdu's static and dynamic parts need to be included.

The second criterion is e.g. required in a pure post-build configuration scenario, where the frame-layout may not be completed at the time of System with category ECU_EXTRACT creation.

10.3.3 ISignals and ISignalGroups

ISignalIPdus carry their information in form of ISignals or ISignalGroups. In order to include all required ISignal and ISignalGroup elements, the following selection criteria apply:

- The System with category ECU_EXTRACT shall contain ISignal elements which are referenced by included ISignalIPdu's ISignalToIPduMapping. One exception are Pdu Gateways. Signal definitions that are not directly relevant for Gateways in case that the Pdu is routed as a whole (Pdu Routing) shall be omitted. See Section 10.3.5 for more details.
- The System with category ECU_EXTRACT shall contain all ISignal elements which are referenced by any included ISignalTriggering.
- The System with category ECU_EXTRACT shall contain ISignalGroup elements which are referenced by included ISignalIPdu's ISignalToIPduMapping. One exception are Pdu Gateways. Signal Group definitions that are not directly relevant for Gateways in case that the Pdu is routed as a whole (Pdu Routing) shall be omitted. See Section 10.3.5 for more details.
- The System with category ECU_EXTRACT shall contain all ISignalGroup elements which are referenced by any included ISignalTriggering.

Like in the case of the Pdu inclusion rules, the second and fourth criterion is required in scenarios with incomplete Pdu modeling due to post-build configurability of the communication matrix.

10.3.4 SystemSignal and SystemSignalGroup

Whereas the rules specified in Section 10.2.3 for the inclusion of SystemSignal comprise all SystemSignals that are being used by the Software Components in the ECU, the inclusion rules above stated for ISignalIPdus and ISignals may require the inclusion of additional SystemSignals. Also, strictly speaking both SystemSignals and SystemSignalGroup need to be considered. The complete inclusion rules for SystemSignals and SystemSignalGroups are:

- The System with category ECU_EXTRACT shall contain all SystemSignals and SystemSignalGroup elements which are referenced by any included DataMapping.
- The System with category ECU_EXTRACT shall contain all SystemSignal elements which are referenced by any included ISignal.
- The System with category ECU_EXTRACT shall contain all SystemSignalGroup elements which are referenced by any included ISignalGroup.

In addition on the receiving EcuInstance the following cases exist:

- only one SystemSignal out of the transmitted SystemSignalGroup is received: no SystemSignalGroup is required in the Ecu Extract of the receiving EcuInstance.
- more than one but not all SystemSignals out of the transmitted SystemSignalGroup are received: new SystemSignalGroup shall be created in the System with category ECU_EXTRACT of the receiving EcuInstance containing the received SystemSignals.
- all SystemSignals out of the transmitted SystemSignalGroup are received: the original SystemSignalGroup shall be the taken over to the System with category ECU_EXTRACT of the receiving EcuInstance.

10.3.5 Gateways

Gateways that refer the EcuInstance shall be included in the System with category ECU_EXTRACT. The complete inclusion rules for Gateways are:

- The System with category ECU_EXTRACT shall contain all FrameMapping elements that are aggregated by the Gateway element.
- The System with category ECU_EXTRACT shall contain all IPduMapping elements that are aggregated by the Gateway element.
- The System with category ECU_EXTRACT shall contain all ISignalMapping elements that are aggregated by the Gateway element.
- ISignal definitions that are not directly relevant for the Gateway in case that the Pdu containing these ISignals is routed as a whole (Pdu Routing) shall be omitted .
- ISignalGroup definitions that are not directly relevant for the Gateways in case that the Pdu containing these ISignalGroups is routed as a whole (Pdu Routing) shall be omitted .

10.3.6 TP configuration

The TP-configuration element TpConfig and all its associated elements shall be included into the System with category ECU_EXTRACT if the EcuInstance has an TpAddress configured in this TpConfig.

10.3.7 NM configuration

The NM-configuration element NmConfig and all its associated elements shall be included into the System with category ECU_EXTRACT if the EcuInstance is included in the NmConfig's configured list of NmEcus.

10.4 Naming Issues

Please note that a detailed information about mechanisms to resolve naming conflicts is given in [4]: [TR_METH_03005], [TR_METH_03006], [TR_METH_03007], [TR_METH_03008], [TR_METH_03009], [TR_METH_03010].

10.4.1 Package Structure

As detailed in the sections above, extracting information from the System with category SYSTEM_DESCRIPTION into an System with category ECU_EXTRACT is a non-trivial transformation: While some of the model elements are simply copied verbatim into the System with category ECU_EXTRACT, it is additionally necessary to create new elements reducing parts of system-wide structures, most noticeably in flattening of the hierarchical VFB view to the ECU Flat View.

All such elements being created or modified in the process of generating the System with category ECU_EXTRACT shall reside in the same ARPackage. In order to avoid namespace conflicts with existing elements, the package shall exclusively be used for this purpose.

By creating derivation elements from elements originally contained in the System with category SYSTEM_DESCRIPTION package structure, duplications of names may occur. This kind of name clashes shall be resolved by a suitable naming algorithm (see section 10.4.3).

All Elements that are taken over from the System with category SYSTEM_DESCRIPTION unchanged (e.g. AtomicSwComponentType, PortInterface, ApplicationDataType, EcuInstance, CommunicationCluster) shall remain in their original packages.

ARElements not used in the System with category ECU_EXTRACT shall not be copied to the ECU Extract XML file.

In more detail, ARPackages taken over from System with category SYSTEM_DESCRIPTION will not be altered by the ECU extraction process, except that some ARElements will not be included in the actual XML file of the extract: ARElements which exist in the System with category SYSTEM_DESCRIPTION but have been stripped for the System with category ECU_EXTRACT are not actually deleted from their ARPackage, but merely are skipped in the XML file forming the extract. Note that having such a partial view on an ARPackage doesn't break the original ARPackage definition because the composition of PackageableElement, responsible for adding ARElements to ARPackage, is stereotyped ≪splitable≫; this means several XML files can contribute to an ARPackage, or in case of the ECU Extract an AUTOSAR description file may contain only a subset of the complete ARPackage.

10.4.2 Naming of Measurement and Calibration Data

The software component descriptions provide several means to declare data prototypes which have to be available for measurement and calibration (MCD) tools on the EcuInstance. Together with the System with category ECU_EXTRACT it is required to provide a list of references to the description of these data for further processing in the scope of the EcuInstance. In addition, the MCD tools need a unique name for each instance of such a data prototype. Since the data descriptions are part of the nested composition structure and are contained in reusable types (components or port interfaces), the system description itself does in general not provide unique names for those.

This means, providing such a list with references and unique names for MCD data is also a task of the ECU extractor tool. This list is part of the artifact ECU Flat Map, which is further explained below.

10.4.3 Naming of Derived Elements

When performing the extract process, name clashes may occur, necessitating a naming scheme for elements derived in ECU generation: By flattening the Software Composition hierarchy all component instances present on the considered EcuInstance are put in one ECU-wide software composition. Name clashes may occur for the following reasons:

1. SwComponentPrototypes taken from different Software Compositions are allowed to have identical short names in the hierarchical structure. As all SwComponentPrototypes will be located in the same ECU Flat View, the original name spaces separation no longer exists.
2. Multiple instances of the same CompositionSwComponentType are mapped to an EcuInstance: In this case, duplicates of all contained SwComponentPrototypes will be placed next to each other in the ECU flat composition.
3. The two mechanisms just mentioned may also lead to name clashes in AutosarDataPrototypes if their names shall be used as MCD data names. In addition, reuse of a PortInterface can also lead to name clashes if it provides data elements to be used by MCD.
4. The setup of PortGroups in the ECU flat view can result in name clashes, because two port groups originating from different component types (i.e. different name spaces) may be aggregated within the flat view.

Therefore the System with category ECU_EXTRACT generator shall take care that all elements derived or created during the extraction process have unique short names. These unique names shall be created in an initial step of the extraction process which leads to the creation of an initial ECU Flat Map. Some ways to satisfy this requirement may be:

- Use globally unique identifiers (GUID) for generating short names.
- Add a number to the original name; if done consistently the flat map approach makes this reproducable.
- Expand the name recursively by the names of the containing elements (e.g. compositions) until is is unique.
- Allow human interaction (this may be combined with an initially proposed name expansion).

The creation of a new short name is compulsory only if otherwise a clash would occur.

[constr_2025] Uniqueness of symbol attributes ⌈ In the context of a single EcuInstance, the values of the RunnableEntity.symbol in combination with the attribute AtomicSwComponentType.symbol of all deployed RunnableEntitys shall be unique such that no two (or more) combinations of RunnableEntity.symbol and AtomicSwComponentType.symbol share the same value. ⌋

10.4.4 Re-use of short names assigned in previous iterations

As described in the previous section, potential name clashes during ECU extraction must be avoided by assigning unique names to the elements specifically created for the System with category ECU_EXTRACT and for the list of MCD data per EcuInstance. Considering the use case of iterative development (also see Section 10.5), the same names shall be assigned to existing elements in consecutive iterations. Elements which have been modified or newly introduced between two ECU extract iterations shall not use an existing short name. Additionally, the ECU extractor tool shall not re-use any short name used in any iteration from previous development phases if the meaning of the element is not exactly the same (i.e. the element's back reference into the System Configuration Description is not the same.)

10.5 ECU Extract in subsequent Cycles of Iterative Development

10.5.1 Traceability of model elements created in ECU Extract

For development scenarios in real life projects iterative development must be supported.

The following use case shall be considered:

Changes in the System with category SYSTEM_DESCRIPTION require the recreation of an System with category ECU_EXTRACT. In the successive re-run of ECU configuration, ECU configuration parameters which were configured based on the previous System with category ECU_EXTRACT need to be maintained for those parts in the System with category ECU_EXTRACT that didn't change between iterations.

Consequently, there are two requirements on the extraction process:

- Elements that are present in both versions of the System with category SYSTEM_DESCRIPTION must not change their short names between the two ECU Extracts either.
- If changes between the two versions of the System with category SYSTEM_DESCRIPTION lead to the creation of new model elements in the System with category ECU_EXTRACT, then these newly created elements must have new names that have not been used in previous iterations of the System with category ECU_EXTRACT. (See also Section 10.4.4).

In order to fulfill these requirements, a back-tracing of the relevant model elements in the System with category ECU_EXTRACT to their counterparts in the System with category SYSTEM_DESCRIPTION shall be established. Based on these back references, short names shall consistently be re-used in iterations. Relevant elements are all those which potentially have been modified in the extraction process.

All back-tracing references are collected in one central table per System with category ECU_EXTRACT based on the meta-class FlatMap. This table collects "instance" entries for each Ecu Extract element that is being created in the System with category ECU_EXTRACT transformation and for each MCD data object that has to be available in the EcuInstance. These entries are called FlatInstanceDescriptor.

Each mapping entry owns two references per mapped element, one reference pointing to the target element in the System with category ECU_EXTRACT, the other one pointing to the origin in the System with category SYSTEM_DESCRIPTION. Both of these references are deep "instance" references, requiring a tuple of context/target description.

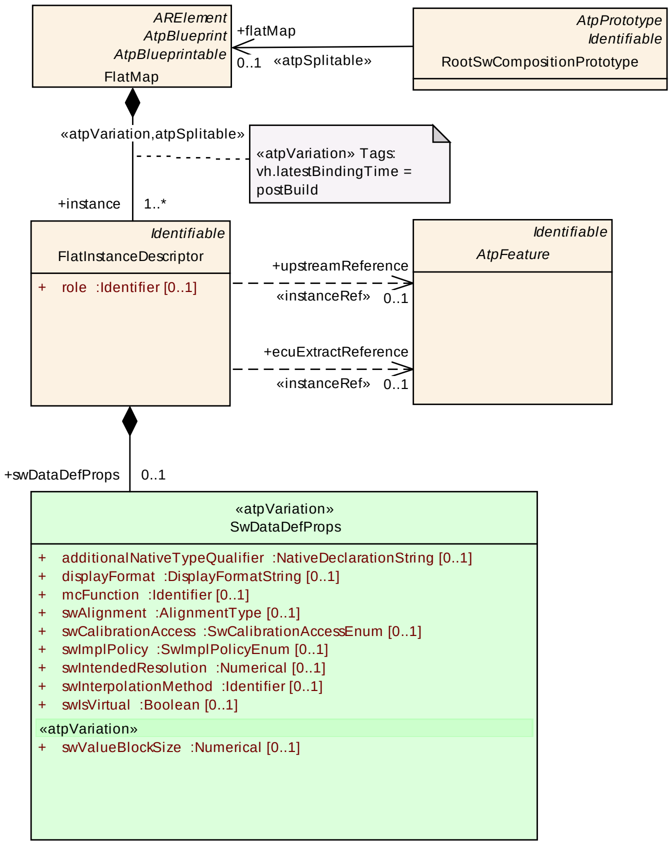

Figure 10.3: Flat Map (CommonStrucure: FlatMap)

Class	**FlatMap**			
Package	M2::AUTOSARTemplates::CommonStructure::FlatMap			
Note	Contains a flat list of references to software objects. This list is used to identify instances and to resolve name conflicts. The scope is given by the RootSwCompositionPrototype for which it is used, i.e. it can be applied to a system, system extract or ECU-extract. An instance of FlatMap may also be used in a preliminary context, e.g. in the scope of a software component before integration into a system. In this case it is not referred by a RootSwCompositionPrototype. **Tags:** atp.recommendedPackage=FlatMaps			
Base	ARElement,ARObject,AtpBlueprint,AtpBlueprintable,Collectable Element,Identifiable,MultilanguageReferrable,PackageableElement,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***

Attribute	Datatype	Mul.	Kind	Note
instance	FlatInstanceDescriptor	1..*	aggr	A descriptor instance aggregated in the flat map. The variation point accounts for the fact, that the system in scope can be subject to variability, and thus the existence of some instances is variable. The aggregation has been made splitable because the content might be contributed by different stakeholders at different times in the workflow. Plus, the overall size might be so big that eventually it becomes more manageable if it is distributed over several files. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=postBuild

Table 10.1: FlatMap

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	Datatype	Mul.	Kind	Note

Attribute	Datatype	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 AtomicSoftwareComponentType, which is refered by the particular SwcInternalBehavior. **Tags:** xml.sequenceOffset=40
role	Identifier	0..1	ref	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. ModeDeclarationGroupPrototypes which are measurable. In this case the RTE will provide locations for current mode, previous mode and next mode.
swDataDef Props	SwDataDefProps	0..1	aggr	The properties of this FlatInstanceDescriptor.
upstreamR eference	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 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 the instance of the component prototype that contains the particular instance of SwcInternalBehavior. **Tags:** xml.sequenceOffset=20

Table 10.2: FlatInstanceDescriptor

[TPS_SYST_01000] FlatInstanceDescriptor roles ⌈ If a ModeDeclarationGroupPrototype is measurable the FlatMap shall contain three entries where the particular roles are set to

- CURRENT_MODE specifies the FlatInstanceDescriptor applicable for current mode value of the ModeDeclarationGroupPrototype
- PREVIOUS_MODE specifies the FlatInstanceDescriptor applicable for previous mode value of the ModeDeclarationGroupPrototype
- NEXT_MODE specifies the FlatInstanceDescriptor applicable for next mode value of the ModeDeclarationGroupPrototype

Please note that these entries may exist in a FlatMap even if the ModeDeclarationGroupPrototype is not measurable. ⌋*(RS_SYST_00003, RS_SYST_00027)*

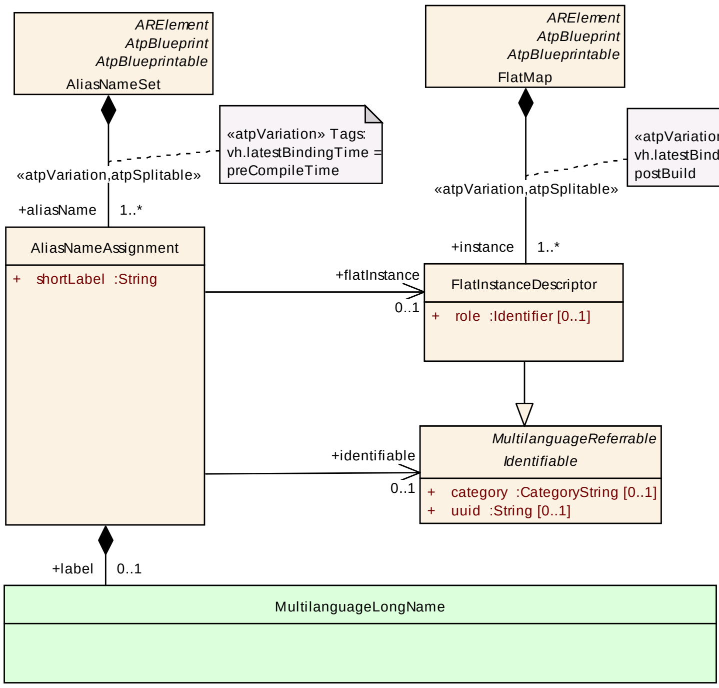

Figure 10.4: Alias Name Assignment (CommonStrucure: AliasNameAssignment)

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. It shall not be used by the RTE generator to generate the MC-Support. In a given instance of AliasNameSet in the bound system there must be at most one aliasName per FlatInstanceDescriptor. **Tags:** atp.recommendedPackage=AliasNameSets			
Base	ARElement,ARObject,AtpBlueprint,AtpBlueprintable,Collectable Element,Identifiable,MultilanguageReferrable,PackageableElement,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**

Attribute	Datatype	Mul.	Kind	Note
aliasName	AliasNameAssignment	1..*	aggr	AliasNames contained in the AliasNameSet. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortLabel vh.latestBindingTime=preCompileTime

Table 10.3: AliasNameSet

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	*Datatype*	*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	MultilanguageLongName	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 10.4: AliasNameAssignment

During the ECU extraction process, the ECU `FlatMap` will be processed in the following steps:

1. Create the entries `shortName` and `upstreamReference` of the `FlatMap` or, if a previous version exists, try to reuse them. Resolve name conflicts.
2. Generate the ECU Software Composition.
3. Create the entries `ecuExtractReference` of the ECU `FlatMap`.

More details are define be the AUTOSAR methodology, see [4]. The methodology also allows to have a `FlatMap` for the whole system. This System `FlatMap` can be created and maintained independently from the ECU extraction process, but can be used as an input for the creation of the ECU FlatMap.

10.5.2 Mapping of AUTOSAR attributes to ASAM ASAP2

With the MC Support information AUTOSAR builds a bridge to tools processing ASAM ASAP2 files. In order to support the interoperability of converter tools the following mapping of AUTOSAR attributes to ASAM ASAP2 [21] (also known as "A2l" respectively "ASAM MCD 2MC") is recommended:

- If the FlatInstanceDescriptor references DataPrototypes:

  FlatInstanceDescriptor.shortName ->
  MEASUREMENT Name
  CHARACTERISTIC Name

  FlatInstanceDescriptor.(longName + desc |upstreamReference.desc) ->
  MEASUREMENT LongIdentifier
  CHARACTERISTIC LongIdentifier

  AliasNameAssignment.shortLabel ->
  MEASUREMENT [-> DISPLAY_IDENTIFIER]
  CHARACTERISTIC [-> DISPLAY_IDENTIFIER]

  AliasNameAssignment.label (if provided) +
  FlatInstanceDescriptor.(desc |upstreamReference.desc) ->
  MEASUREMENT LongIdentifier
  CHARACTERISTIC LongIdentifier

- If AliasNameAssignment references a SwSystemconstant:

  AliasNameAssignment.shortLabel ->
  SYSTEM_CONSTANT -> Name for SwSystemconstants

- If AliasNameAssignment references a Unit:

  AliasNameAssignment.shortLabel ->
  UNIT -> Name for Units

10.6 Variant Handling in ECU Extract

The System Template supports the creation of variants in many of its model elements. Depending on the binding time, some of this variability may have been already resolved within the System with category SYSTEM_DESCRIPTION at the time of creating the System with category ECU_EXTRACT, and a cleanup step may have removed some of the complexity by removing the out-configured variability.

If however binding of a concrete variation condition happens in a later stage of the AUTOSAR methodology (e.g. during ECU Configuration or even post build), or if for other process reasons such a cleanup step is not applicable, the variability needs to be carried over to the System with category ECU_EXTRACT.

10.6.1 System Constants

In the AUTOSAR variant handling concept, SwSystemconst represents a variant selector which needs to have its value assigned latest at binding time of any expression which refers to it. Such a value assignment may be done literally using a fixed value, or by specifying a formula, depending on the values of other variant selectors. The elements to do this are collected in a SwSystemconstantValueSet, aggregating individual value assignment expressions in the form of SwSystemconstValue.

In the System with category ECU_EXTRACT, all SwSystemconst elements are included that influence its variable content. In detail the following rules for the inclusion of SwSystemconst apply:

- System with category ECU_EXTRACT shall contain all SwSystemconst elements that are being referenced directly by variable elements contained in the System with category ECU_EXTRACT.
- Additionally, whenever a SwSystemconst's value is assigned indirectly using an SwSystemconstValue's ConditionByFormula expression, each SwSystemconstValue referred to in the assignment formula needs to be included, too. As such assignments may be nested in multiple levels, the whole directed acyclic graph of SwSystemconst elements influencing the System with category ECU_EXTRACT variability need to be included.

Additionally to the SwSystemconst elements also all relevant SwSystemconstValue assignments need to be included. As they are aggregated by SwSystemconstantValueSet, the whole Value Set is included whenever one of its SwSystemconstValue assignments is relevant for the System with category ECU_EXTRACT.

Note: Typically, the assignment of Variants ("Binding´´) will be done in a Variant Configuration work product, separate from the actual System with category ECU_EXTRACT. In this case, the relevant information from the Variant Configuration also needs to be extracted and delivered in combination with the System with category ECU_EXTRACT. From the model point of view it doesn't matter whether System with category ECU_EXTRACT and Variant Configuration are contained in the same file or in separate files.

10.6.2 Nested Whole/Part class variants

In case of flattening the hierarchical VFB view to the ECU flat view representation, the case may appear that one conditional SwComponentPrototype is nested within another SwComponentPrototype depending on another variance condition. As the resulting ECU flat view only has a flat representation of SwComponentPrototypes, such a double condition needs to be resolved to a single condition in the resulting SwComponentPrototypes.

In this case, the variation condition formula needs to be altered such that the two (or more) individual conditions are combined in a boolean AND function.

A Glossary

Artifact This is a Work Product Definition that provides a description and definition for tangible work product types. Artifacts may be composed of other artifacts ([22]).

At a high level, an artifact is represented as a single conceptual file.

AUTOSAR Tool This is a software tool which supports one or more tasks defined as AUTOSAR tasks in the methodology. Depending on the supported tasks, an AUTOSAR tool can act as an authoring tool, a converter tool, a processor tool or as a combination of those (see separate definitions).

AUTOSAR Authoring Tool An AUTOSAR Tool used to create and modify AUTOSAR XML Descriptions. Example: System Description Editor.

AUTOSAR Converter Tool An AUTOSAR Tool used to create AUTOSAR XML files by converting information from other AUTOSAR XML files. Example: ECU Flattener

AUTOSAR Definition This is the definition of parameters which can have values. One could say that the parameter values are Instances of the definitions. But in the meta model hierarchy of AUTOSAR, definitions are also instances of the meta model and therefore considered as a description. Examples for AUTOSAR definitions are: `EcucParameterDef`, `PostBuildVariantCriterion`, `SwSystemconst`.

AUTOSAR XML Description In AUTOSAR this means "filled Template". In fact an AUTOSAR XML description is the XML representation of an AUTOSAR model.

The AUTOSAR XML description can consist of several files. Each individual file represents an AUTOSAR partial model and shall validate successfully against the AUTOSAR XML schema.

AUTOSAR Meta-Model This is an UML2.0 model that defines the language for describing AUTOSAR systems. The AUTOSAR meta-model is an UML representation of the AUTOSAR templates. UML2.0 class diagrams are used to describe the attributes and their interrelationships. Stereotypes, UML tags and OCL expressions (object constraint language) are used for defining specific semantics and constraints.

AUTOSAR Model This is a representation of an AUTOSAR product. The AUTOSAR model represents aspects suitable to the intended use according to the AUTOSAR methodology.

Strictly speaking, this is an instance of the AUTOSAR meta-model. The information contained in the AUTOSAR model can be anything that is representable according to the AUTOSAR meta-model.

AUTOSAR Partial Model In AUTOSAR, the possible partitioning of models is marked in the meta-model by ≪`atpSplitable`≫. One partial model is represented in an AUTOSAR XML description by one file. The partial model does not need to fulfill all semantic constraints applicable to an AUTOSAR model.

AUTOSAR Processor Tool An AUTOSAR Tool used to create non-AUTOSAR files by processing information from AUTOSAR XML files. Example: RTE Generator

AUTOSAR Template The term "Template" is used in AUTOSAR to describe the format different kinds of descriptions. The term template comes from the idea, that AUTOSAR defines a kind of form which shall be filled out in order to describe a model. The filled form is then called the description.

In fact the AUTOSAR templates are now defined as a meta model.

AUTOSAR XML Schema This is a W3C XML schema that defines the language for exchanging AUTOSAR models. This Schema is derived from the AUTOSAR meta model. The AUTOSAR XML Schema defines the AUTOSAR data exchange format.

Blueprint This is a model from which other models can be derived by copy and refinement. Note that in contrast to meta model resp. types, this process is *not* an instantiation.

Instance Generally this is a particular exemplar of a model or of a type.

Life Cycle Life Cycle is the course of development/evolutionary stages of a model element during its life time.

Meta-Model This defines the building blocks of a model. In that sense, a Meta-Model represents the language for building models.

Meta-Data This includes pertinent information about data, including information about the authorship, versioning, access-rights, timestamps etc.

Model A Model is an simplified representation of reality. The model represents the aspects suitable for an intended purpose.

Partial Model This is a part of a model which is intended to be persisted in one particular artifact.

Pattern in GST : This is an approach to simplify the definition of the meta model by applying a model transformation. This transformation creates an enhanced model out of an annotated model.

Property A property is a structural feature of an object. As an example a “connector” has the properties “receive port” and “send port”

Properties are made variant by the ≪atpVariation≫.

Prototype This is the implementation of a role of a type within the definition of another type. In other words a type may contain Prototypes that in turn are typed by "Types". Each one of these prototypes becomes an instance when this type is instantiated.

Type A type provides features that can appear in various roles of this type.

Value This is a particular value assigned to a “Definition”.

Variability Variability of a system is its quality to describe a set of variants. These variants are characterized by variant specific property settings and / or selections. As an example, such a system property selection manifests itself in a particular "receive port" for a connection.

This is implemented using the ≪atpVariation≫.

Variant A system variant is a concrete realization of a system, so that all its properties have been set respectively selected. The software system has no variability anymore with respect to the binding time.

This is implemented using EvaluatedVariantSet.

Variation Binding A variant is the result of a variation binding process that resolves the variability of the system by assigning particular values/selections to all the system's properties.

This is implemented by VariationPoint.

Variation Binding Time The variation binding time determines the step in the methodology at which the variability given by a set of variable properties is resolved.

This is implemented by vh.LatestBindingtime at the related properties .

Variation Definition Time The variation definition time determines the step in the methodology at which the variation points are defined.

Variation Point A variation point indicates that a property is subject to variation. Furthermore, it is associated with a condition and a binding time which define the system context for the selection / setting of a concrete variant.

This is implemented by VariationPoint.

B Supported special use-cases

The description means of the communication matrix in the System Template potentially support a variety of use-cases. Some combinations of description means are explicitly ruled-out by semantical constraints. But the remaining space for the possible descriptions is so huge, that certain use-cases are actually not supported by tool-vendors because they did not consider them. This chapter describes special use-cases that can be specified in the System Template in order to get a harmonized support by tools.

B.1 Support of sending / receiving same Can/Flexray Frame on same channel (Pdu Gateway Use-Case)

Description: The System Template supports the definition of a communication where the same Can/Flexray FrameTriggering is sent and received on the same PhysicalChannel of one Pdu Gateway EcuInstance.

Rationale: This use-case occurs in gateway EcuInstances which are used in several vehicle platforms.

Implementation: This usage shall be supported by defining one Frame and one FrameTriggering with different directions on the referenced FramePorts for the same PhysicalChannel. Also one Pdu and one PduTriggering with different directions on the referenced IPduPorts for the same PhysicalChannel shall be used.

Example: In figure B.1 a sample network setup is shown. The ECU1 is designed to send the Frame_X on the PhysicalChannel. The ECU2, ECU3 and ECU4 do receive the information. But since ECU1 is optional, ECU4 is also designed to send the Frame_X on the network (in case ECU1 is not present). Please note that in in this example ECU4 is a gateway EcuInstance that is connected to an additional channel.

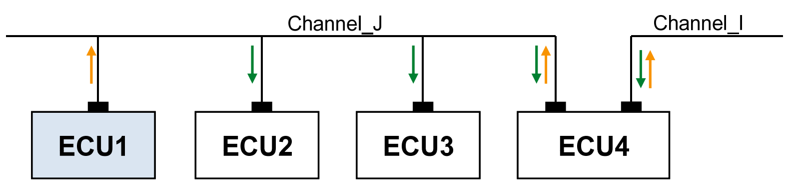

Figure B.1: Example of network setup with one Frame being received and sent on the same ECU and channel

In the system description there exists one definition for the Frame_X and one FrameTriggering for the PhysicalChannel (figure B.2). Each EcuInstance sending or receiving the FrameTriggering does define one FramePort per direction, thus for ECU4 there are two FramePorts defined.

For each Pdu mapped to the Frame there exists one definition for the Pdu_X and one PduTriggering for the PhysicalChannel. Each EcuInstance sending or receiving the Pdu does define one IPduPort per direction, thus for ECU4 there are two IPduPorts defined.

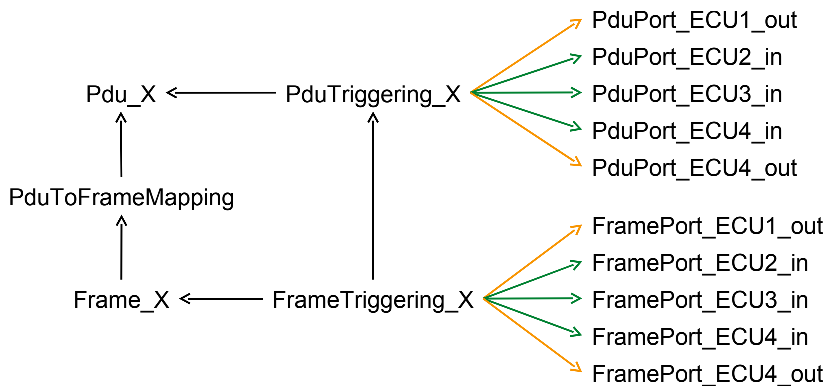

Figure B.2: Structure to reflect the frame- and pdu-triggering setup of one Frame being received and sent by the same Gateway ECU

In case a System Extract / ECU Extract is build, only the relevant FramePorts and IPduPorts for the corresponding EcuInstance are extracted. Especially in case an additional EcuInstance is designed to send and receive the same Frame all the other ECU extracts will not be affected by this change.

B.2 Support of sending / receiving same Can/Flexray Frame on same channel (bidirectional routing in COM)

Description: The System Template supports the definition of a communication where the same Can/Flexray FrameTriggering is sent and received on the same PhysicalChannel of one EcuInstance and the content of this Frame is processed by an Application.

Rationale: This use-case occurs in case of runtime variation where the same data is transmitted or received by the same ECU.

Implementation in a System Description: This use-case is supported with the following modelling:

- One Frame and one FrameTriggering with different directions on the referenced FramePorts for the same PhysicalChannel shall be defined.
- One Pdu and one PduTriggering with different directions on the referenced IPduPorts for the same PhysicalChannel shall be defined.
- One ISignal and one ISignalTriggering with different directions on the referenced ISignalPorts for the same PhysicalChannel shall be defined.

Please note that in case of a bidirectional routing on the ISignal level the COM Configuration (ComIPdus) needs to be derived from the PduTriggering and from IPduPorts.

Example: In figure B.3 a sample network setup is shown. The same data (Frame_X) is transmitted by Ecu4 and by Ecu1 (runtime variation). Ecu4 is designed to send and to receive the Frame_X on the network. For Ecu2 and Ecu3 it is transparent from which sender (Ecu1 or Ecu4) the data is transmitted.

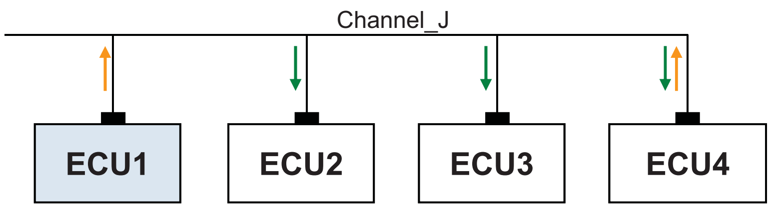

Figure B.3: Example of network setup with one Frame being received and sent on the same ECU and channel

In the system description there exists one definition for the Frame_X and one FrameTriggering for the PhysicalChannel (figure B.4). Each EcuInstance sending or receiving the FrameTriggering does define one FramePort per direction, thus for ECU4 there are two FramePorts defined.

For each Pdu mapped to the Frame there exists one definition for the Pdu_X and one PduTriggering for the PhysicalChannel. Each EcuInstance sending or receiving the Pdu does define one IPduPort per direction, thus for ECU4 there are two IPduPorts defined.

For each ISignal mapped to the Pdu there exists one definition for the Signal_X and one ISignalTriggering for the PhysicalChannel. Each EcuInstance sending or receiving the ISignal does define one ISignalPort per direction, thus for ECU4 there are two ISignalPorts defined.

Example B.4 shows a System Description where only the DataMapping for the PPorts is defined. Please note that in the COM configuration a ComIPdu has a ComIPduDirection. Therefore two ComIPdus (Tx and Rx) need to be created from such a System Description.

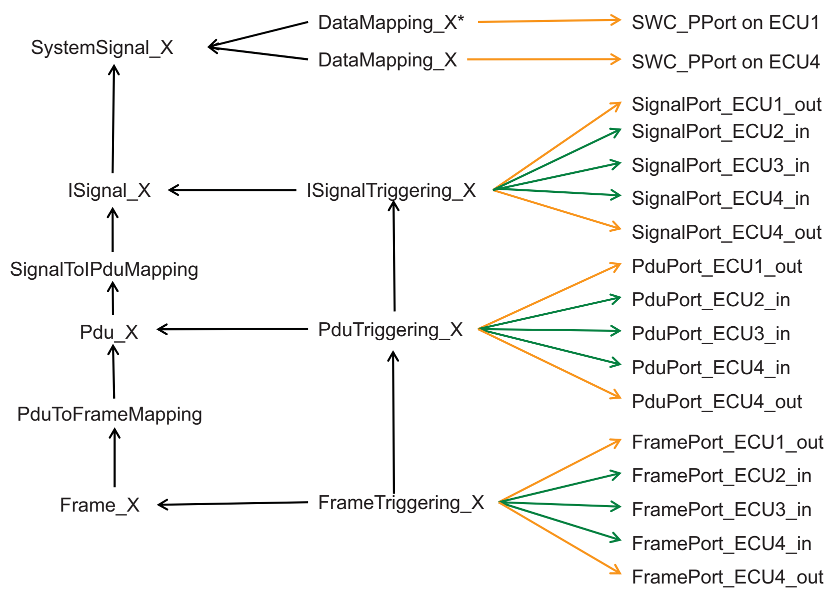

Figure B.4: Structure to reflect the frame- and pdu-triggering setup of one Frame being received and sent on the same ECU and channel (System Description with ECU1, ECU2, ECU3 and ECU4)

In case a System Extract / ECU Extract is build, only the relevant FramePorts, IPduPorts and ISignalPorts for the corresponding EcuInstance are extracted. Especially in case an additional EcuInstance is designed to send and receive the same Frame all the other ECU extracts will not be affected by this change. Figure B.5 shows a System Extract where only the description for ECU4 is available. Please note that in this example the VariableDataPrototype in the PPort and the VariableDataPrototype in the RPort of the Software Component are mapped to the same SystemSignal.

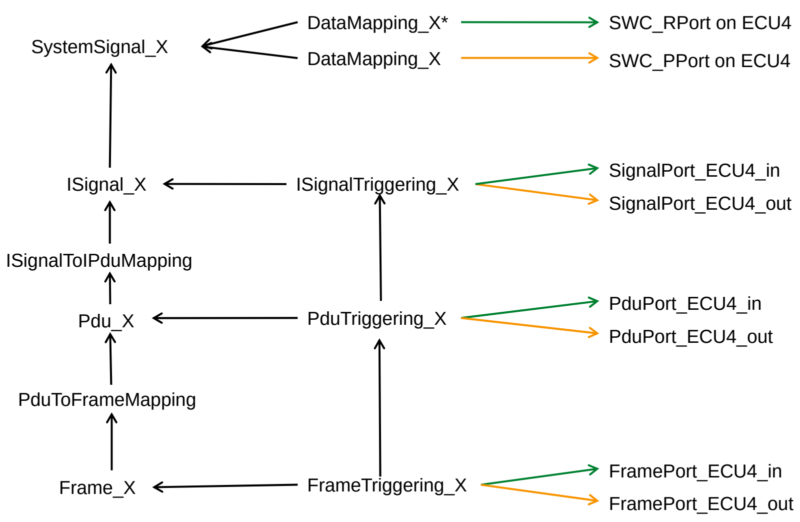

Figure B.5: Structure to reflect the frame- and pdu-triggering setup of one Frame being received and sent on the same ECU and channel (System Extract with ECU4 only)

B.3 Support of Frame, Pdus and Signals with length 0

The AUTOSAR client-server communication requires to support signals with length zero. If no actual data is configured for a client-server communication, i. e. the applicable ClientServerToSignalGroupMapping owns only an emptySignal, the RTE sends a signal group with an emptySignal to initiate the communication. In this case the element EmptySignalMapping in the ClientServerToSignalGroupMapping shall reference a SystemSignal that is referenced by an ISignal with length equal to zero. Such empty ISignals will be mapped into Pdus and Frames and therefore Pdus and Frame with length zero are also supported by the System Template.

B.4 Support of dynamic CAN IDs

To support efficient diagnostics with on-board clients, efficient routing, and efficient SAE J1939 transport protocol and request handling, AUTOSAR provides access to dynamic CAN identifier parts in upper layers of the COM stack. This is achieved by appending parts of the identifier (or the complete identifier) as MetaData to the Pdu payload. The System Template uses three attributes for the configuration of this feature:

- The metaDataLength of a Pdu defines the number of bytes reserved in the payload of the Pdu for MetaData. Please note that the length and metaDataLength define together the size of the Pdu.
- The rxMask of a CanFrameTriggering defines the relevant bits in a CAN identifier and thus defines a range of CAN identifiers that match these bits and may vary in the other bits.
- The txMask of a CanFrameTriggering defines the static bits in a CAN identifier and thus allows to set the other bits using the data appended to the payload.

These three parameters are sufficient to support the following scenarios:

- A Pdu is transmitted from one AUTOSAR node to another with variable ID parts. In this case, rxMask and txMask will be identical, and the variable identifier parts placed in the Pdu MetaData by the sender will be routed transparently and received in the same way.
- A Pdu is transmitted by one node with a static identifier and received using the rxMask. In this case, the MetaData is not used, and the receiver is tolerant regarding dynamic address parts.
- J1939 Pdu is sent with fixed priority, but priority is ignored by the receiver. Here, the MetaData may or may not be used, and the rxMask differs from the txMask just in the three priority bits.

B.5 Description of MOST Functions

The MOST communication protocol is not supported by the AUTOSAR Basic Software but it is possible to convert FIBEX [9] descriptions with MOST content to an AUTOSAR description. This chapter describes how MOST Functions may be described with the means of the Software Component Template [5].

FIBEX supports the description of SW-PACKAGES (represents a bundle of FBlocks and implemented MOST functions), MOST-FUNCTION-BLOCKs (contain functions with operation types and finally parameters, e.g. CD Player), MOST-FUNCTIONs (e.g. a CD player possesses functions such as Play, Stop, Eject, and Time Played) and OP-TYPEs (operations that are applied to the respective function, e.g. Play.Set(tracknumber)). The following table shows how the FIBEX elements may be converted into an AUTOSAR description.

MOST FIBEX Element	*Description*	*AUTOSAR Element*	*Mapping Rule*
FUNCTION-BLOCK	A MOST device contains multiple components that are called function blocks, for example, tuner, amplifier, or CD player.	SwComponentType	Each FunctionBlock shall be described as a SwComponentType
FUNCTION-BLOCK-INSTANCE	There may be several Instances with the same FBlockID in the system (two CD changers, four active speakers, several diagnosis blocks)	SwComponentPrototype	Each FunctionBlock-Instance shall be described as a SwComponentPrototype
MOST-FUNCTION	Methods and Properties of a Function Block (e.g. Play, Stop...)	ClientServerInterface	Methods and Properties shall be described as ClientServerInterfaces
OP-TYPE	The OPType indicates which operation must be applied to the property or method (e.g. Play.Start, Property.Get)	ClientServerOperation	Methods and Properties shall be described as ClientServerOperations.

MOST FIBEX Element	*Description*	*AUTOSAR Element*	*Mapping Rule*
OP-TYPE Parameter	Parameters of OP-TYPE (e.g. track-number)	ArgumentDataPrototype	OP-TYPE Parameters shall be described as ArgumentDataPrototypes of ClientServerOperations.
CLUSTER (MOST-Cluster)	MOST CommunicationCluster	UserDefinedCluster	A MOST CommunicationCluster shall be described as UserDefinedCluster that allows the modeling of arbitrary Communication Clusters. A MOST-Cluster may aggregate several PhysicalChannels
CHANNEL	The CHANNEL object is used to specify the communications channel used by individual OPTypes.	UserDefinedPhysicalChannel	A UserDefinedPhysicalChannel shall be described for each CHANNEL (Control Channel and/or a MOST High Protocol) that is used by the MOST CommunicationCluster.
PDU TRIGGERING	The PDU-TRIGGERING is created for every OP-TYPE that is transported on this CHANNEL.	PduTriggering	A PduTriggering shall be created for every Pdu that contains ClientServerOperations that correspond to a OP-Type and shall be transported on the PhysicalChannel that aggregates this PduTriggering.

MOST FIBEX Element	*Description*	*AUTOSAR Element*	*Mapping Rule*
PDU	In FIBEX the OP-TYPE corresponds to a PDU in the communication description	Pdu	In AUTOSAR the ClientServerOperation representing the OP-TYPE shall be mapped with the ClientServerToSignalMapping to a SystemSignal. For the SystemSignal an ISignal shall be created. The ISignal is mapped into an ISignalIPdu.

Table B.1: FIBEX to AUTOSAR MOST Element Mapping

C Detailed Representation of InstanceRef Associations in the System Template

As a special type of association "instanceRef" refers to an exact instance of the referenced class, requiring additional information of the target and the context. This is explained in detail in the AUTOSAR Generic Structure Template [2]. This chapter contains the detailed InstanceRef Diagrams.

C.1 Usage of InstanceRefs in Data Mapping diagrams

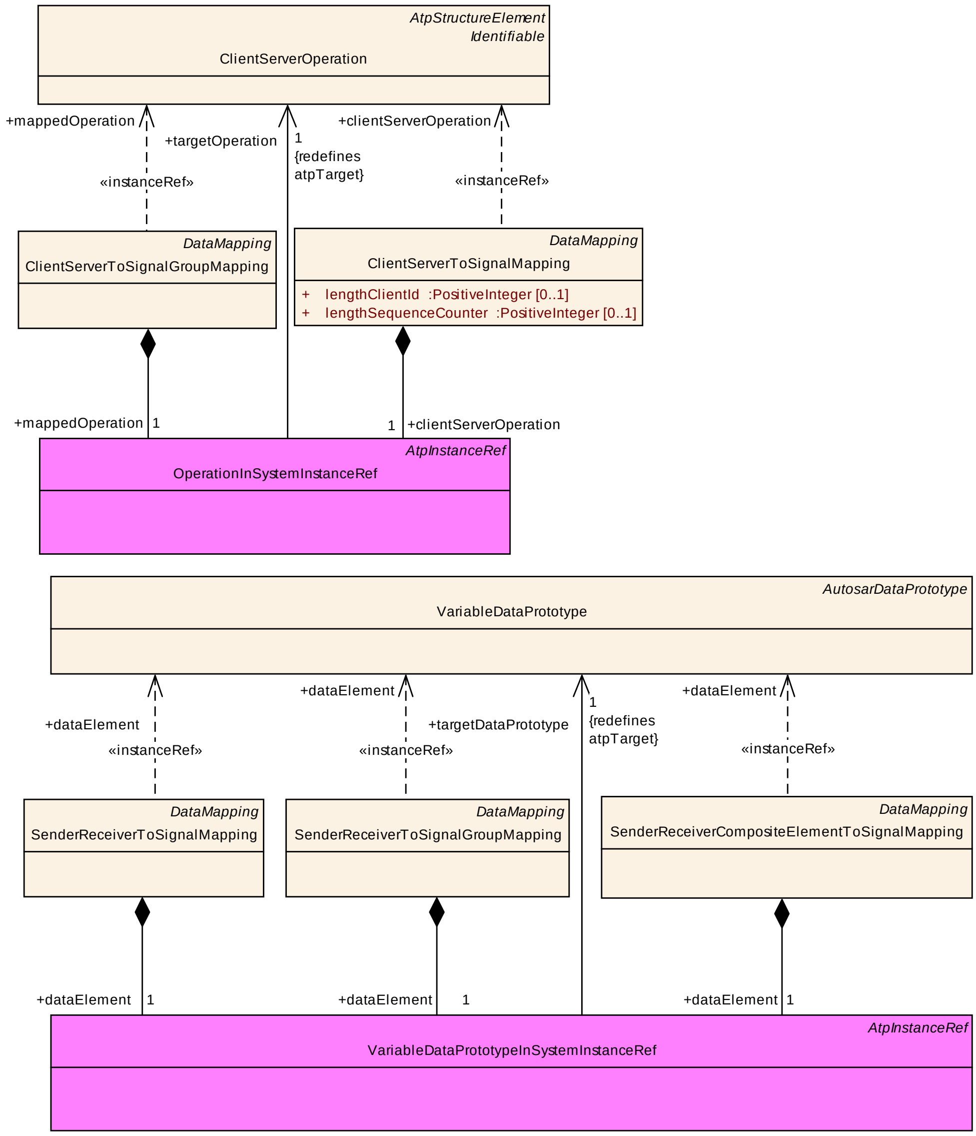

Figure C.1: Data Mapping Instance Ref Usage

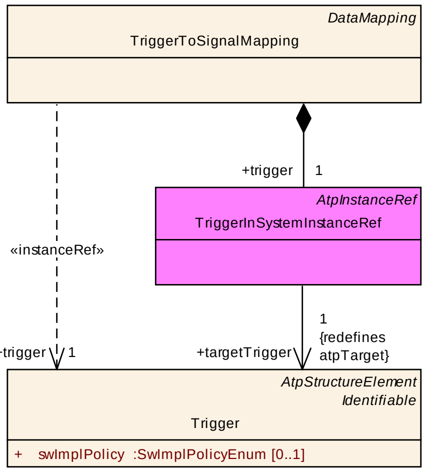

Figure C.2: Modeling of InstanceRef usage for `TriggerInSystemInstanceRef`

C.2 Usage of InstanceRefs in SW Mapping diagrams

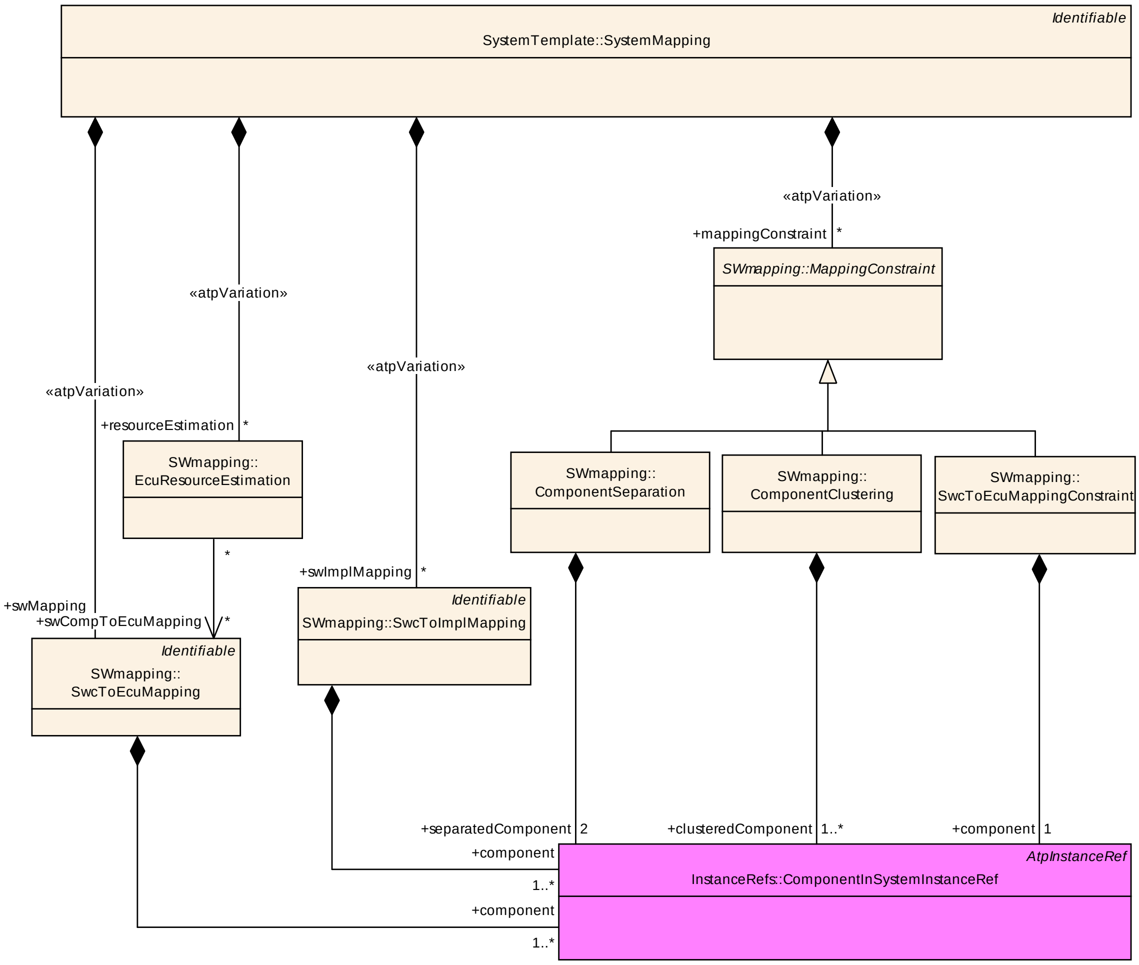

Figure C.3: SW Mapping Instance Ref Usage

C.3 Usage of InstanceRefs in Signal Path Constraint diagrams

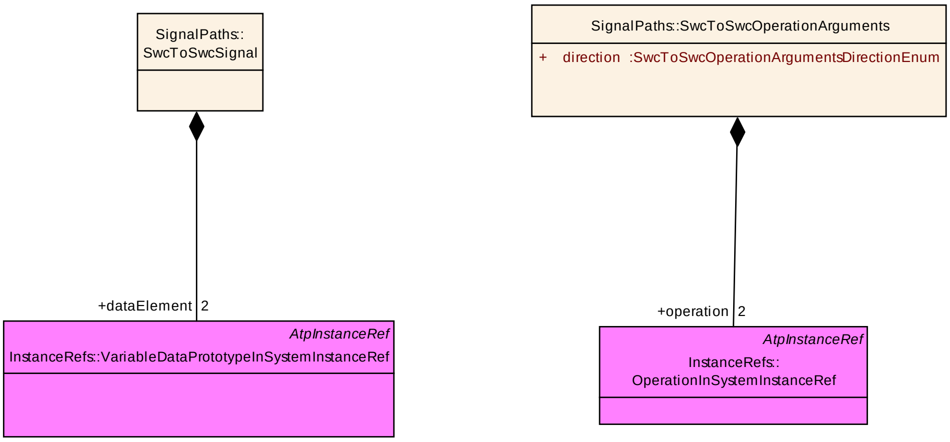

Figure C.4: SW Mapping Instance Ref Usage

C.4 Usage of InstanceRefs in PncMapping

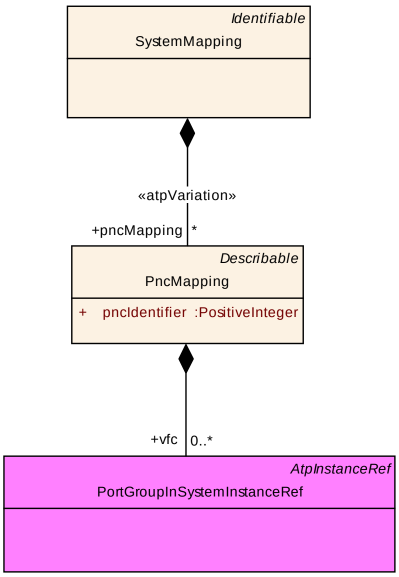

Figure C.5: Partial Network Mapping Instance Ref Usage

C.5 "SWC in System" InstanceRef

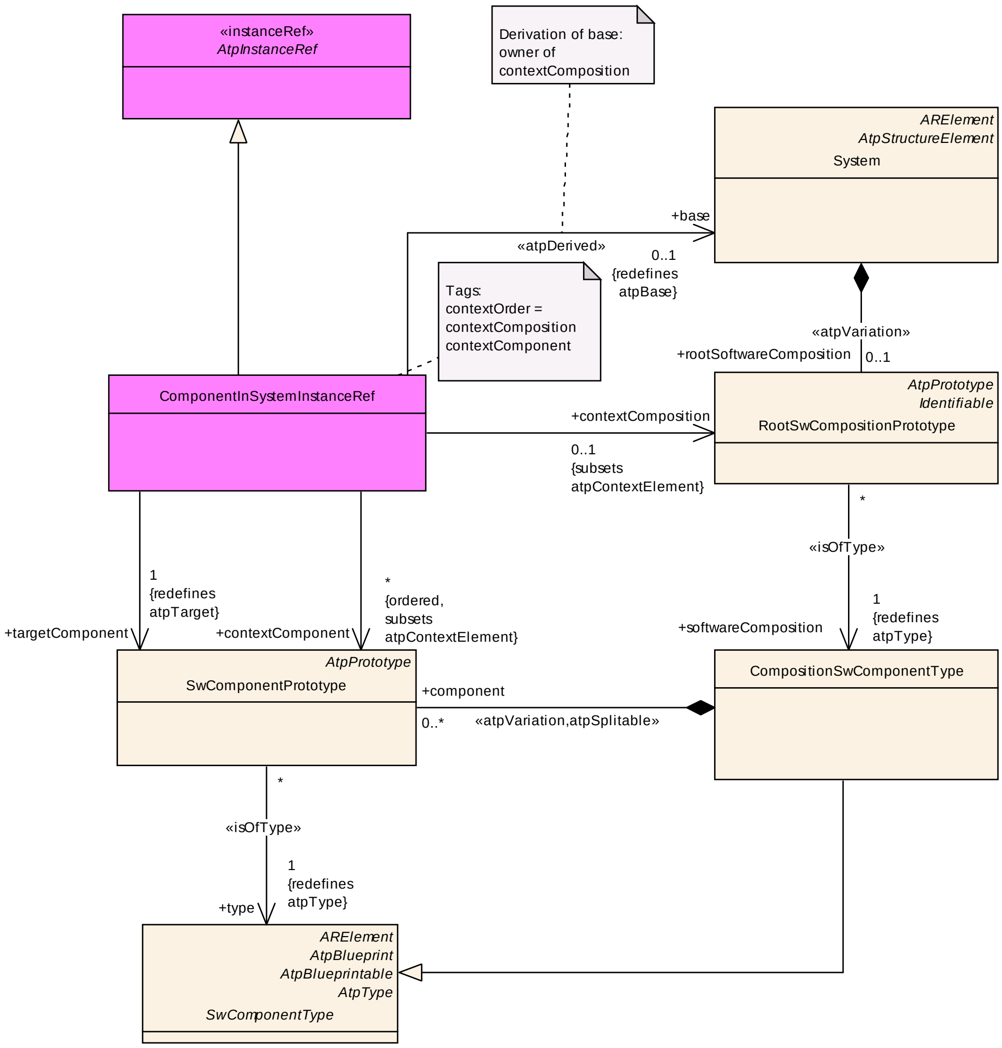

Figure C.6: ComponentInSystem InstanceRef

Class	**ComponentInSystemInstanceRef**			
Package	M2::AUTOSARTemplates::SystemTemplate::InstanceRefs			
Note				
Base	ARObject,AtpInstanceRef			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
base	System	0..1	ref	**Stereotypes:** atpDerived**Tags:** xml.sequenceOffset=10
contextComponent (ordered)	SwComponentPrototype	*	ref	**Tags:** xml.sequenceOffset=30
contextComposition	RootSwCompositionPrototype	0..1	ref	**Tags:** xml.sequenceOffset=20
targetComponent	SwComponentPrototype	1	ref	**Tags:** xml.sequenceOffset=40

Table C.1: ComponentInSystemInstanceRef

If the referenced SwComponentPrototype is located within the RootSwCompositionPrototype of a System then the base and the contextComposition to the RootSwCompositionPrototype shall be provided. If the referenced SwComponentPrototype is the RootSwCompositionPrototype itself then the base reference and the contextComposition reference to the RootSwCompositionPrototype shall be skipped and only the targetComponent to the RootSwCompositionPrototype shall be used.

C.6 "Operation in System" InstanceRef

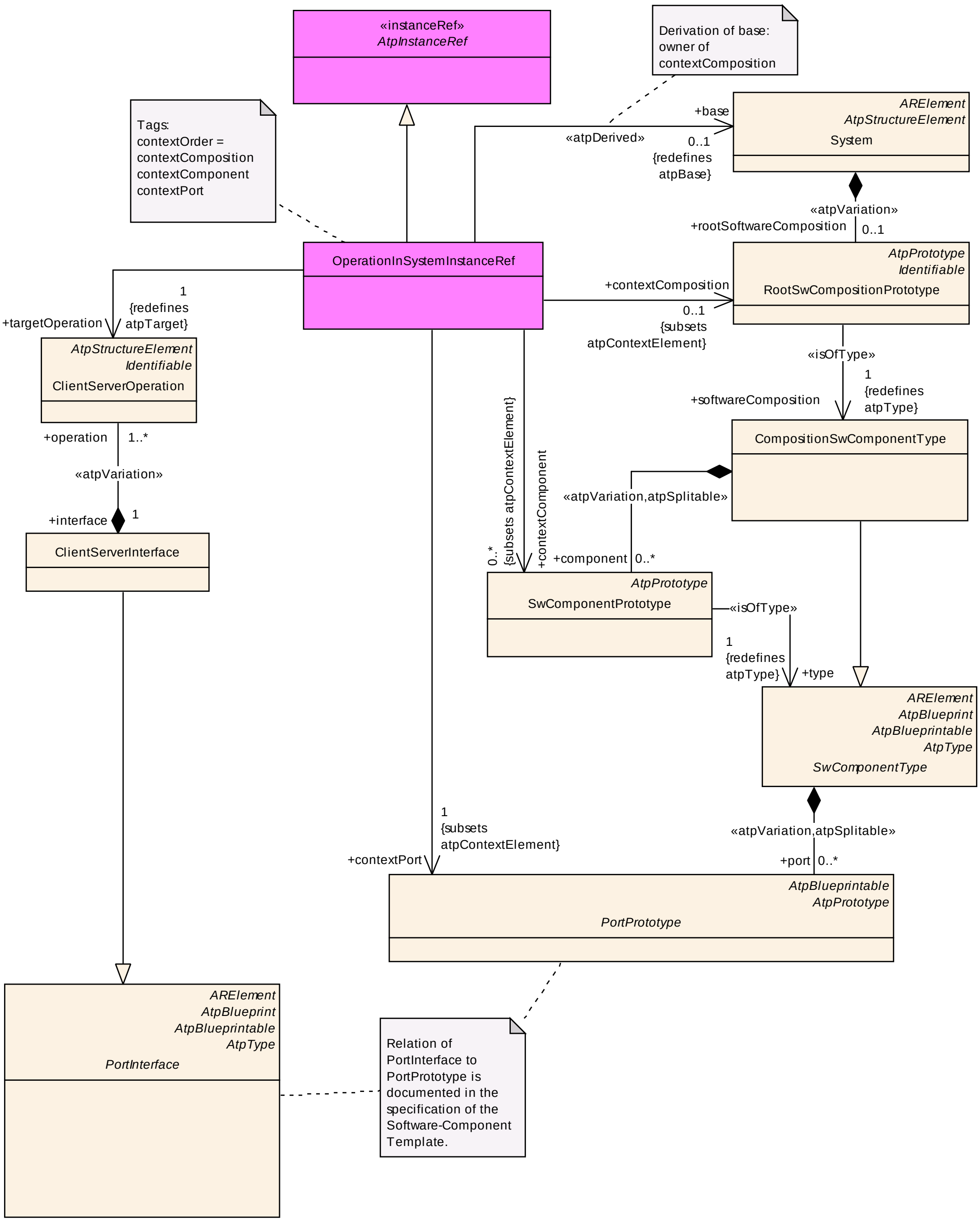

Figure C.7: OperationInSystem InstanceRef

Class	**OperationInSystemInstanceRef**			
Package	M2::AUTOSARTemplates::SystemTemplate::InstanceRefs			
Note				
Base	ARObject,AtpInstanceRef			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***

Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
base	System	0..1	ref	**Stereotypes:** atpDerived**Tags:** xml.sequenceOffset=10
contextComponent	SwComponentPrototype	*	ref	**Tags:** xml.sequenceOffset=30
contextComposition	RootSwCompositionPrototype	0..1	ref	**Tags:** xml.sequenceOffset=20
contextPort	PortPrototype	1	ref	**Tags:** xml.sequenceOffset=40
targetOperation	ClientServerOperation	1	ref	**Tags:** xml.sequenceOffset=50

Table C.2: OperationInSystemInstanceRef

If the referenced `ClientServerOperation` is part of a `PortInterface` of a `SwComponentPrototype` that is located within the `RootSwCompositionPrototype` then the `base` reference and the `contextComposition` reference to the `RootSwCompositionPrototype` shall be provided. If the referenced `ClientServerOperation` is part of a `PortInterface` of the `RootSwCompositionPrototype` itself then the `base` reference and the `contextComposition` reference to the `RootSwCompositionPrototype` shall be skipped and the `RootSwCompositionPrototype` shall be referenced as `contextComponent`.

C.7 "VariableDataPrototype" InstanceRef

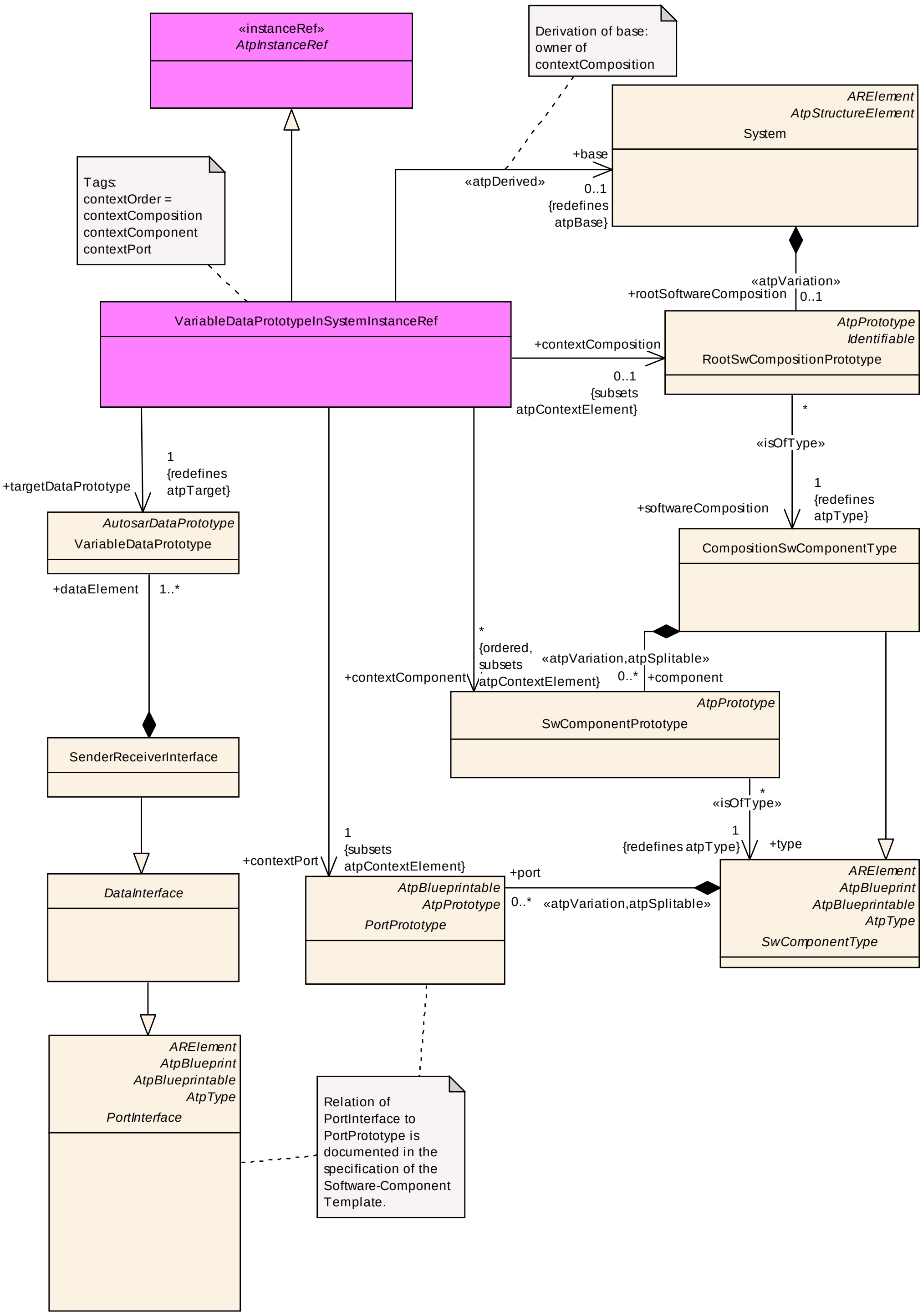

Figure C.8: VariableDataPrototypeInSystem InstanceRef

Class	VariableDataPrototypeInSystemInstanceRef			
Package	M2::AUTOSARTemplates::SystemTemplate::InstanceRefs			
Note				
Base	ARObject,AtpInstanceRef			
Attribute	Datatype	Mul.	Kind	Note
base	System	0..1	ref	**Stereotypes:** atpDerived
contextComponent (ordered)	SwComponentPrototype	*	ref	
contextComposition	RootSwCompositionPrototype	0..1	ref	
contextPort	PortPrototype	1	ref	
targetDataPrototype	VariableDataPrototype	1	ref	

Table C.3: VariableDataPrototypeInSystemInstanceRef

If the referenced VariableDataPrototype is part of a PortInterface of a SwComponentPrototype that is located within the RootSwCompositionPrototype then the base reference and the contextComposition reference to the RootSwCompositionPrototype shall be provided. If the referenced VariableDataPrototype is part of a PortInterface of the RootSwCompositionPrototype itself then the base reference and the contextComposition reference to the RootSwCompositionPrototype shall be skipped and the RootSwCompositionPrototype shall be referenced as contextComponent.

Please note that the xml.sequenceOffset is not set for this InstanceRef and therefore the properties are serialized in an alphabetical order.

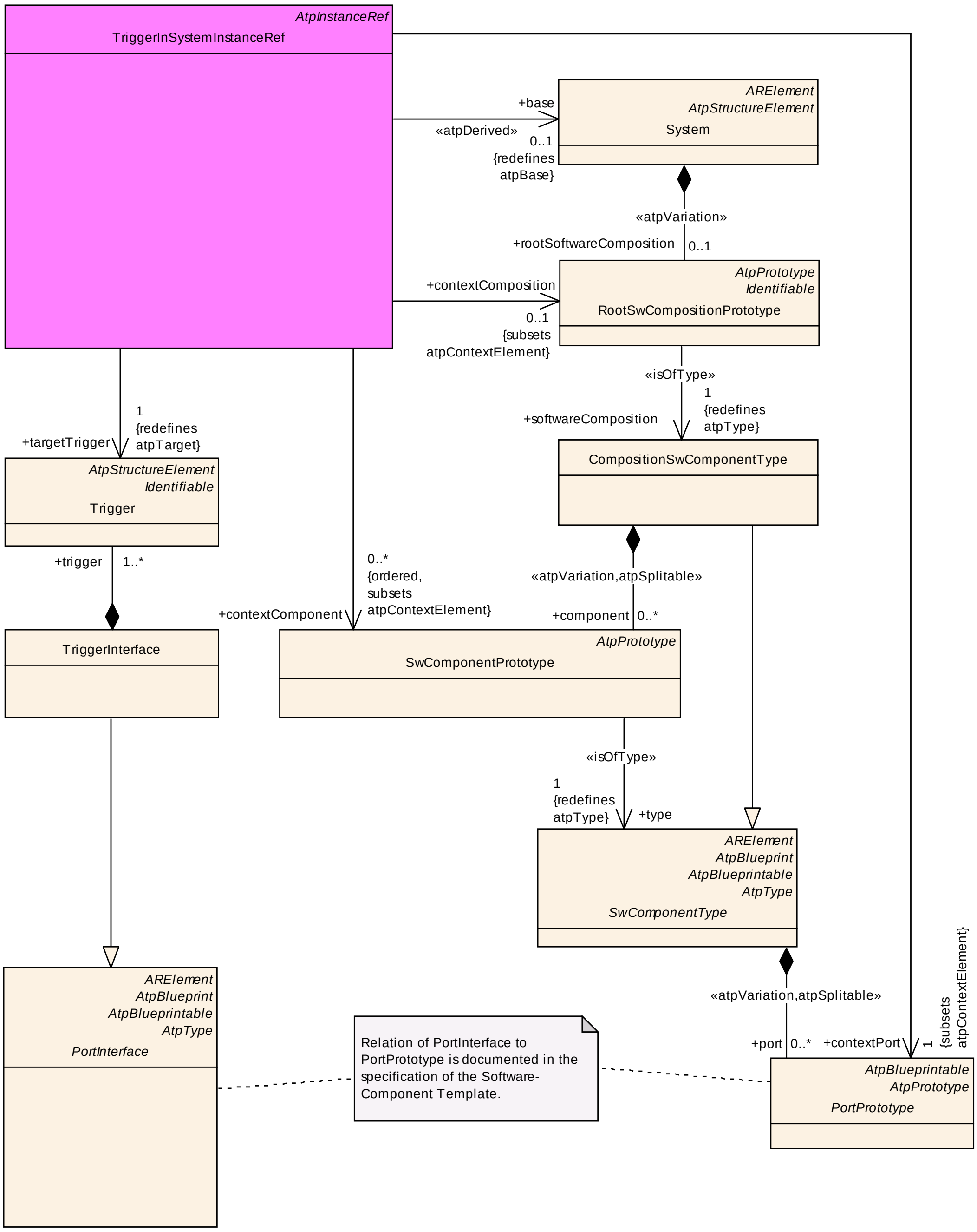

Figure C.9: TriggerInSystemInstanceRef

Class	TriggerInSystemInstanceRef			
Package	M2::AUTOSARTemplates::SystemTemplate::InstanceRefs			
Note				
Base	ARObject,AtpInstanceRef			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
base	System	0..1	ref	This represents that base of the InstanceRef **Stereotypes:** atpDerived **Tags:** xml.sequenceOffset=10

Attribute	Datatype	Mul.	Kind	Note
contextComponent (ordered)	SwComponentPrototype	*	ref	This represents the set of context components. The association is ordered because it needs to respect the nesting order. **Tags:** xml.sequenceOffset=30
contextComposition	RootSwCompositionPrototype	0..1	ref	This represents tha reference to the RootSwCompositiontype representing a context of the InstanceRef. **Tags:** xml.sequenceOffset=20
contextPort	PortPrototype	1	ref	This represents the PortPrototype in which the target Trigger is located. **Tags:** xml.sequenceOffset=40
targetTrigger	Trigger	1	ref	This represents the target Trigger. **Tags:** xml.sequenceOffset=50

Table C.4: TriggerInSystemInstanceRef

If the referenced `Trigger` is part of a `PortInterface` of a `SwComponentPrototype` that is located within the `RootSwCompositionPrototype` then the `base` reference and the `contextComposition` reference to the `RootSwCompositionPrototype` shall be provided. If the referenced `Trigger` is part of a `PortInterface` of the `RootSwCompositionPrototype` itself then the `base` reference and the `contextComposition` reference to the `RootSwCompositionPrototype` shall be skipped and the `RootSwCompositionPrototype` shall be referenced as `contextComponent`.

C.8 "PortGroup in System" InstanceRef

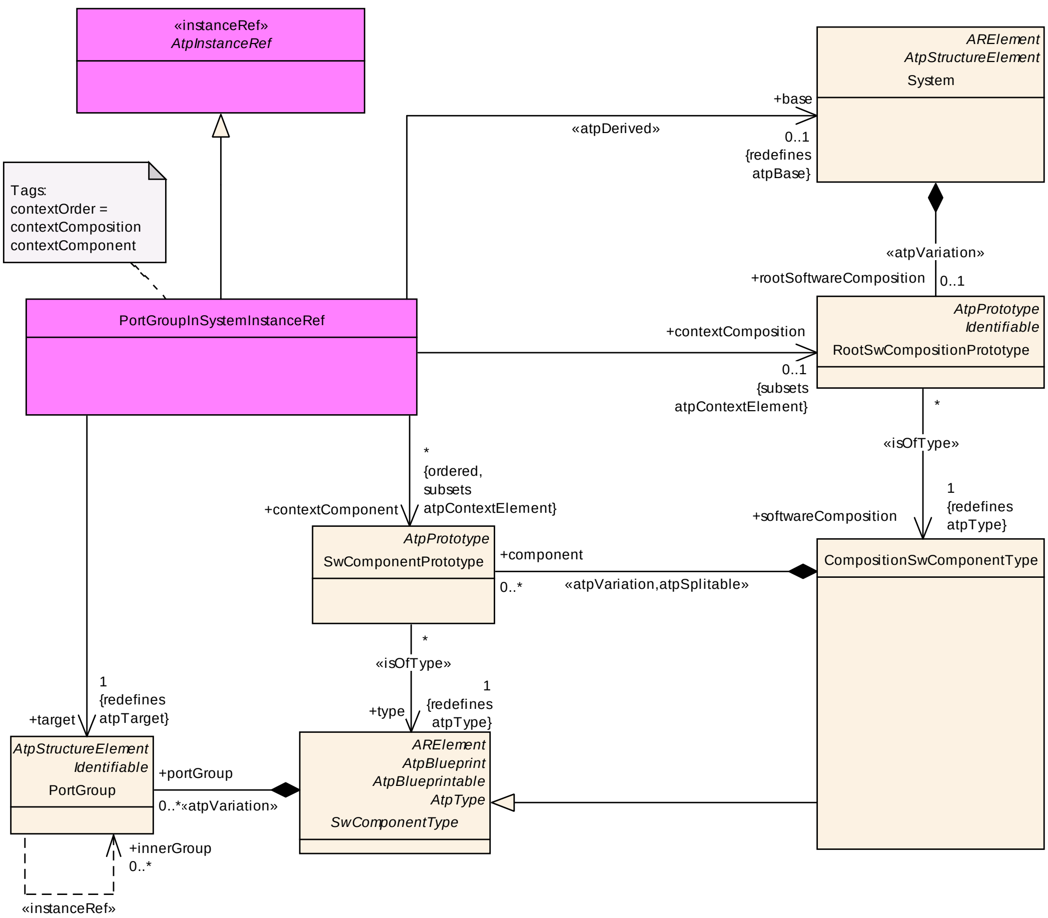

Figure C.10: PortGroupInSystem InstanceRef

Class	**PortGroupInSystemInstanceRef**			
Package	M2::AUTOSARTemplates::SystemTemplate::InstanceRefs			
Note				
Base	ARObject,AtpInstanceRef			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
base	System	0..1	ref	**Stereotypes:** atpDerived**Tags:** xml.sequence Offset=10
contextComponent (ordered)	SwComponentPrototype	*	ref	**Tags:** xml.sequenceOffset=30
contextComposition	RootSwCompositionPrototype	0..1	ref	**Tags:** xml.sequenceOffset=20
target	PortGroup	1	ref	Link to a PortGroup that is defined in a component which is part of this CompositionSwComponentType. **Tags:** xml.sequenceOffset=40

Table C.5: PortGroupInSystemInstanceRef

If the referenced PortGroup is part of a SwComponentPrototype that is located within the RootSwCompositionPrototype then the base and the contextComposition reference to the RootSwCompositionPrototype shall be provided. If the referenced PortGroup is part of the RootSwCompositionPrototype itself then the base and the contextComposition reference to the RootSwCompositionPrototype shall be skipped and the RootSwCompositionPrototype shall be referenced as contextComponent.

D Harmonisation between Upstream Templates and ECU Configuration

This chapter describes the mapping of the ECU Configuration parameters (M1 model) onto the meta-classes and attributes of the AUTOSAR upstream templates (System Template, SW Component Template and ECU Resource Template).

The relationships between upstream templates and ECU Configuration are described in order to answer typical questions like:

- How shall a supplier use the information in a System Description in order to fulfill the needs defined by the systems engineer?
- How is a tool vendor supposed to generate an ECU Configuration Description out of ECU Extract of System Description?

Please note that the tables contain the following columns:

bsw module: Name of BSW module

bsw context: Reference to parameter container

bsw type: Type of parameter

bsw param: Name of the BSW parameter

bsw desc: Description from the configuration document

m2 template: System Template, SW Component Template, ECU Resource Template

m2 param: Name of the upstream template parameter

m2 description: Description from the upstream template definition

mapping rule: Textual description on how to transform between M2 and BSW domains

mapping type:

- local: no mapping needed since parameter local to BSW
- partial: some data can be automatically mapped but not all
- full: all data can be automatically mapped

D.1 ComStack

D.1.1 Com Mapping

BSW Module	BSW Context	
Com	Com	
BSW Parameter		**BSW Type**
ComConfig		EcucParamConfContainerDef
BSW Description		
This container contains the configuration parameters and sub containers of the AUTOSAR COM module. This container is a MultipleConfigurationContainer, i.e. this container and its sub-containers exist once per configuration set.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Com	Com/ComConfig	
BSW Parameter		**BSW Type**
ComConfigurationId		EcucIntegerParamDef
BSW Description		
This ID is returned by a call to Com_GetConfigurationId.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Com	Com/ComConfig	
BSW Parameter		**BSW Type**
ComDataMemSize		EcucIntegerParamDef
BSW Description		
Size of internal Com data in units of bytes (static memory allocation) - memory required by post-build configuration must be smaller than this constant. This parameter is needed only in case of post-build loadable implementation using static memory allocation.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
Com	Com/ComConfig	
BSW Parameter		**BSW Type**

ComGwMapping		EcucParamConfContainerDef	
BSW Description			
Each instance of this container defines one mapping of the integrated Signal Gateway.			
M2 Template	**M2 Description**		
TPS_SYST	Arranges those signals that are transferred by the gateway from one channel to the other in pairs and defines the mapping between them.		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Multiplatform::Gateway::SignalMapping			
Mapping Rule			**Mapping Type**
Create Container for each ISignalMapping that is defined in the ECU Extract.			full

BSW Module	**BSW Context**		
Com	Com/ComConfig/ComGwMapping		
BSW Parameter		**BSW Type**	
ComGwDestination		EcucChoiceContainerDef	
BSW Description			
Each instance of this choice container allows to define one routing destination either by reference to an already configured COM signal / signal group or by a destination description container.			
M2 Template	**M2 Description**		
TPS_SYST	Target destination of the referencing mapping.		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Multiplatform::Gateway::SignalMapping.targetSignal			
Mapping Rule			**Mapping Type**
Create Container for each targetSignal reference that is defined in the ISignal Mapping.			full

BSW Module	**BSW Context**		
Com	Com/ComConfig/ComGwMapping/ComGwDestination		
BSW Parameter		**BSW Type**	
ComGwDestinationDescription		EcucParamConfContainerDef	
BSW Description			
Description of a gateway destination. This container allows defining a gateway destination without the configuration of a complete COM signal. This allows adding / changing gateway relations post build without the configuration of new signals.			
M2 Template	**M2 Description**		
TPS_SYST	An ISignalToIPduMapping describes the mapping of ISignals to SignalIPdus and defines the position of the ISignal within an SignalIPdu.		
M2 Parameter			
Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping			
Mapping Rule			**Mapping Type**
Informations can be derived from ISignalToIPduMapping			full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestination Description	
BSW Parameter		**BSW Type**
ComBitPosition		EcucIntegerParamDef
BSW Description		
Starting position within the I-PDU. This parameter refers to the position in the I-PDU and not in the shadow buffer. If the endianness conversion is configured to Opaque the parameter ComBitPosition shall define the bit0 of the first byte like in little endian byte order		

M2 Template	**M2 Description**	
TPS_SYST	This parameter is necessary to describe the bitposition of a signal within an SignalIPdu.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping.startPosition		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestination Description	
BSW Parameter		**BSW Type**
ComFilter		EcucParamConfContainerDef
BSW Description		
This container contains the configuration parameters of the AUTOSAR COM module's Filters. Note: On sender side the container is used to specify the transmission mode conditions.		
M2 Template	**M2 Description**	
TPS_SWCT, TPS_SYST	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.	
M2 Parameter		
CommonStructure::Filter::DataFilter		
Mapping Rule		**Mapping Type**
Create container on the receiver side if the NonqueuedReceiverComSpec contains a DataFilter. Create Container on the sender side if the TransmissionMode Condition element contains a reference to this signal.		full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestination Description/ComFilter	
BSW Parameter		**BSW Type**
ComFilterAlgorithm		EcucEnumerationParamDef
BSW Description		
The range of values is specified in the [17] specification, chapter 2.2.2, Reception Filtering.		
M2 Template	**M2 Description**	
TPS_SWCT, TPS_SYST	This attriburte specifies the type of the filter.	
M2 Parameter		
CommonStructure::Filter::DataFilter.dataFilterType		
Mapping Rule		**Mapping Type**
Mapping between DataFilterTypeEnum and ComFilterAlgorithm Enum is necessary.		full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestination Description/ComFilter	
BSW Parameter		**BSW Type**
ComFilterMask		EcucIntegerParamDef
BSW Description		

The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering.		
M2 Template	**M2 Description**	
TPS_SWCT, TPS_SYST	Mask for old and new value.	
M2 Parameter		
CommonStructure::Filter::DataFilter.mask		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestination Description/ComFilter	
BSW Parameter		**BSW Type**
ComFilterMax		EcucIntegerParamDef
BSW Description		
The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering.		
M2 Template	**M2 Description**	
TPS_SWCT, TPS_SYST	Value to specify the upper boundary	
M2 Parameter		
CommonStructure::Filter::DataFilter.max		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestination Description/ComFilter	
BSW Parameter		**BSW Type**
ComFilterMin		EcucIntegerParamDef
BSW Description		
The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering.		
M2 Template	**M2 Description**	
TPS_SWCT, TPS_SYST	Value to specify the lower boundary	
M2 Parameter		
CommonStructure::Filter::DataFilter.min		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestination Description/ComFilter	
BSW Parameter		**BSW Type**
ComFilterOffset		EcucIntegerParamDef
BSW Description		

The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering. Range = 0..(ComFilterPeriod-1)		
M2 Template	**M2 Description**	
TPS_SWCT, TPS_SYST	Specifies the initial number of messages to occur before the first message is passed	
M2 Parameter		
CommonStructure::Filter::DataFilter.offset		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestination Description/ComFilter	
BSW Parameter		**BSW Type**
ComFilterPeriod		EcucIntegerParamDef
BSW Description		
This parameter defines the period of the ComFilterAlgorithm ONE_EVERY_N.		
M2 Template	**M2 Description**	
TPS_SWCT, TPS_SYST	specifies number of messages to occur before the message is passed again	
M2 Parameter		
CommonStructure::Filter::DataFilter.period		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestination Description/ComFilter	
BSW Parameter		**BSW Type**
ComFilterX		EcucIntegerParamDef
BSW Description		
The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering.		
M2 Template	**M2 Description**	
TPS_SWCT, TPS_SYST	Value to compare with	
M2 Parameter		
CommonStructure::Filter::DataFilter.x		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestination Description	
BSW Parameter		**BSW Type**
ComGwIPduRef		EcucReferenceDef
BSW Description		
Reference to an I-PDU of a Signal Gateway source or destination description.		
M2 Template	**M2 Description**	

TPS_SYST	Definition of SignalToIPduMappings included in the SignalIPdu.	
M2 Parameter		
Fibex::FibexCore::CoreCommunication::SignalIPdu::ISignalToIPduMapping		
Mapping Rule		**Mapping Type**
Create reference for each existing ISignalToIPduMapping that is referenced from the regarded Signal Gateway.		full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestination Description	
BSW Parameter		**BSW Type**
ComSignalEndianness		EcucEnumerationParamDef
BSW Description		
Defines the endianness of the signal's network representation.		
M2 Template	**M2 Description**	
TPS_SYST	This parameter defines the order of the bytes of the signal and the packing into the IPdu. The byte ordering Little Endian (MostSignificantByteLast), Big Endian (MostSignificantByteFirst) and Opaque can be selected.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping.packingByte Order		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestination Description	
BSW Parameter		**BSW Type**
ComSignalInitValue		EcucStringParamDef
BSW Description		
Initial value for this signal. In case of UINT8_N the default value is a string of length ComSignal-Length with all bytes set to 0x00. In case of UINT8_DYN the initial size shall be 0. In case the ComSignalType is UINT8, UINT16, UINT32, SINT8, SINT16, SINT32 the string shall be interpreted as defined in the chapter Integer Type in the AUTOSAR EcuC specification. In case the ComSignalType is FLOAT32, FLOAT64 the string shall be interpreted as defined in the chapter Float Type in the AUTOSAR EcuC specification. In case the ComSignalType is BOOLEAN the string shall be interpreted as defined in the chapter Boolean Type in the AUTOSAR EcuC specification. In case the ComSignal is a UINT8_N, UINT8_DYN the string shall be interpreted as a decimal representation of the characters separated by blanks, e.g. "97 98 100" means a string "abd", where the char "a" is in byte 0(lowest address), "b" is in byte 1, and "d" is in byte 2 and (highest address). For the ComSignalType UINT8_DYN the dynamic length shall be set to the number of configured characters. An empty string "" shall be interpretated as 0-sized dynamic signal.		
M2 Template	**M2 Description**	
TPS_SWCT, TPS_SYST	If a full DataMapping exist this information may be available from a configured SenderComSpec and ReceiverComSpec. In case the System Description doesn't use a complete Software ComponentDescription an optional reference from SystemSignal is used.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignal.initValue,SWComponentTemplate::Communication::NonqueuedSenderComSpec.initValue		

Mapping Rule	Mapping Type
It is possible to aggregate an initValue at the level of a ComSpec in the SW C Template. in case the System Description doesn't use a complete Software Component Description (VFB View) the initValue is defined in the System Template.	full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestination Description	
BSW Parameter		**BSW Type**
ComTransferProperty		EcucEnumerationParamDef
BSW Description		
Defines if a write access to this signal can trigger the transmission of the corresponding I-PDU. If the I-PDU is triggered, depends also on the transmission mode of the corresponding I-PDU.		
M2 Template	**M2 Description**	
TPS_SYST	The triggered transfer property causes immediate transmission of the IPdu, except if transmission mode Periodic or transmission mode NONE is defined for the IPdu. The Pending transfer property does not cause transmission of an I-PDU.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping.transferProperty		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwDestination Description	
BSW Parameter		**BSW Type**
ComUpdateBitPosition		EcucIntegerParamDef
BSW Description		
Bit position of update-bit inside I-PDU. If this attribute is omitted then there is no update-bit. This setting must be consistently on sender and on receiver side. Range: 0..63 for CAN and LIN 0..2031 for FlexRay		
M2 Template	**M2 Description**	
TPS_SYST	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.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping.updateIndication BitPosition		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComGwMapping/ComGwDestination	
BSW Parameter		**BSW Type**
ComGwSignal		EcucParamConfContainerDef

BSW Description	
This container allows specifying a gateway source or destination respectively with a reference to a ComSignal, a ComGroupSignal or a ComSignalGroup.	
M2 Template	**M2 Description**
TPS_SYST	An ISignalToIPduMapping describes the mapping of ISignals to SignalIPdus and defines the position of the ISignal within an SignalIPdu.
M2 Parameter	
Fibex::FibexCore::CoreCommunication::ISignalTriggering.iSignal	
Mapping Rule	**Mapping Type**
Create Container if ISignal is referenced from Gateway::SignalMapping::ISignal Triggering.	full

BSW Module	**BSW Context**
Com	Com/ComConfig/ComGwMapping/ComGwDestination/ComGwSignal
BSW Parameter	**BSW Type**
ComGwSignalRef	EcucChoiceReferenceDef
BSW Description	
Reference to an object of a gateway relation. Either to a ComSignal, ComGroupSignal or to a SignalGroup.	
M2 Template	**M2 Description**
TPS_SYST	An ISignalToIPduMapping describes the mapping of ISignals to SignalIPdus and defines the position of the ISignal within an SignalIPdu.
M2 Parameter	
Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping	
Mapping Rule	**Mapping Type**
System Template element ISingalToIPduMapping represents the ComSignal, ComSignalGroup or ComGroupSignal.	full

BSW Module	**BSW Context**
Com	Com/ComConfig/ComGwMapping
BSW Parameter	**BSW Type**
ComGwSource	EcucChoiceContainerDef
BSW Description	
This choice container allows the definition of the gateway source signal either by reference to an already configured COM signal / signal group or by a source description container.	
M2 Template	**M2 Description**
TPS_SYST	Source destination of the referencing mapping.
M2 Parameter	
SystemTemplate::Fibex::Fibex4Multiplatform::Gateway::SignalMapping.sourceSignal	
Mapping Rule	**Mapping Type**
Create Container for sourceSignal reference that is defined in the ISignalMapping.	full

BSW Module	**BSW Context**
Com	Com/ComConfig/ComGwMapping/ComGwSource
BSW Parameter	**BSW Type**
ComGwSignal	EcucParamConfContainerDef
BSW Description	
This container allows specifying a gateway source or destination respectively with a reference to a ComSignal, a ComGroupSignal or a ComSignalGroup.	
M2 Template	**M2 Description**

<table>
<tr><td>TPS_SYST</td><td colspan="2">An ISignalToIPduMapping describes the mapping of ISignals to SignalIPdus and defines the position of the ISignal within an SignalIPdu.</td></tr>
<tr><td colspan="3">M2 Parameter</td></tr>
<tr><td colspan="3">Fibex::FibexCore::CoreCommunication::ISignalTriggering.iSignal</td></tr>
<tr><td colspan="2">Mapping Rule</td><td>Mapping Type</td></tr>
<tr><td colspan="2">Create Container if ISignal is referenced from Gateway::SignalMapping::ISignal Triggering.</td><td>full</td></tr>
</table>

<table>
<tr><td>BSW Module</td><td colspan="3">BSW Context</td></tr>
<tr><td>Com</td><td colspan="3">Com/ComConfig/ComGwMapping/ComGwSource/ComGwSignal</td></tr>
<tr><td colspan="2">BSW Parameter</td><td colspan="2">BSW Type</td></tr>
<tr><td colspan="2">ComGwSignalRef</td><td colspan="2">EcucChoiceReferenceDef</td></tr>
<tr><td colspan="4">BSW Description</td></tr>
<tr><td colspan="4">Reference to an object of a gateway relation. Either to a ComSignal, ComGroupSignal or to a SignalGroup.</td></tr>
<tr><td>M2 Template</td><td colspan="3">M2 Description</td></tr>
<tr><td>TPS_SYST</td><td colspan="3">An ISignalToIPduMapping describes the mapping of ISignals to SignalIPdus and defines the position of the ISignal within an SignalIPdu.</td></tr>
<tr><td colspan="4">M2 Parameter</td></tr>
<tr><td colspan="4">Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping</td></tr>
<tr><td colspan="3">Mapping Rule</td><td>Mapping Type</td></tr>
<tr><td colspan="3">System Template element ISingalToIPduMapping represents the ComSignal, ComSignalGroup or ComGroupSignal.</td><td>full</td></tr>
</table>

<table>
<tr><td>BSW Module</td><td colspan="3">BSW Context</td></tr>
<tr><td>Com</td><td colspan="3">Com/ComConfig/ComGwMapping/ComGwSource</td></tr>
<tr><td colspan="2">BSW Parameter</td><td colspan="2">BSW Type</td></tr>
<tr><td colspan="2">ComGwSourceDescription</td><td colspan="2">EcucParamConfContainerDef</td></tr>
<tr><td colspan="4">BSW Description</td></tr>
<tr><td colspan="4">Description of a gateway source. This container allows defining a gateway source without the configuration of a complete COM signal. This allows adding / changing gateway relations post build without the configuration of new signals.</td></tr>
<tr><td>M2 Template</td><td colspan="3">M2 Description</td></tr>
<tr><td>TPS_SYST</td><td colspan="3">An ISignalToIPduMapping describes the mapping of ISignals to SignalIPdus and defines the position of the ISignal within an SignalIPdu.</td></tr>
<tr><td colspan="4">M2 Parameter</td></tr>
<tr><td colspan="4">Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping</td></tr>
<tr><td colspan="3">Mapping Rule</td><td>Mapping Type</td></tr>
<tr><td colspan="3">Informations can be derived from ISignalToIPduMapping</td><td>full</td></tr>
</table>

<table>
<tr><td>BSW Module</td><td colspan="2">BSW Context</td></tr>
<tr><td>Com</td><td colspan="2">Com/ComConfig/ComGwMapping/ComGwSource/ComGwSourceDescription</td></tr>
<tr><td colspan="2">BSW Parameter</td><td>BSW Type</td></tr>
<tr><td colspan="2">ComBitPosition</td><td>EcucIntegerParamDef</td></tr>
<tr><td colspan="3">BSW Description</td></tr>
<tr><td colspan="3">Starting position within the I-PDU. This parameter refers to the position in the I-PDU and not in the shadow buffer. If the endianness conversion is configured to Opaque the parameter ComBitPosition shall define the bit0 of the first byte like in little endian byte order</td></tr>
<tr><td>M2 Template</td><td colspan="2">M2 Description</td></tr>
<tr><td>TPS_SYST</td><td colspan="2">This parameter is necessary to describe the bitposition of a signal within an SignalIPdu.</td></tr>
</table>

M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping.startPosition	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	BSW Context	
Com	Com/ComConfig/ComGwMapping/ComGwSource/ComGwSourceDescription	
BSW Parameter		**BSW Type**
ComBitSize		EcucIntegerParamDef
BSW Description		
Size in bits, for integer signal types. For ComSignalType UINT8_N and UINT8_DYN the size shall be configured by ComSignalLength. For ComSignalTypes FLOAT32 and FLOAT64 the size is already defined by the signal type and therefore may be omitted.		
M2 Template	**M2 Description**	
TPS_SYST	Size of the signal in bits.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignal.length		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Com	Com/ComConfig/ComGwMapping/ComGwSource/ComGwSourceDescription	
BSW Parameter		**BSW Type**
ComGwIPduRef		EcucReferenceDef
BSW Description		
Reference to an I-PDU of a Signal Gateway source or destination description.		
M2 Template	**M2 Description**	
TPS_SYST	Definition of SignalToIPduMappings included in the SignalIPdu.	
M2 Parameter		
Fibex::FibexCore::CoreCommunication::SignalIPdu::ISignalToIPduMapping		
Mapping Rule		**Mapping Type**
Create reference for each existing ISignalToIPduMapping that is referenced from the regarded Signal Gateway.		full

BSW Module	BSW Context	
Com	Com/ComConfig/ComGwMapping/ComGwSource/ComGwSourceDescription	
BSW Parameter		**BSW Type**
ComSignalEndianness		EcucEnumerationParamDef
BSW Description		
Defines the endianness of the signal's network representation.		
M2 Template	**M2 Description**	
TPS_SYST	This parameter defines the order of the bytes of the signal and the packing into the IPdu. The byte ordering Little Endian (MostSignificantByteLast), Big Endian (MostSignificantByteFirst) and Opaque can be selected.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping.packingByteOrder		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Com	Com/ComConfig/ComGwMapping/ComGwSource/ComGwSourceDescription	
BSW Parameter		**BSW Type**
ComSignalLength		EcucIntegerParamDef
BSW Description		
Description: For ComSignalType UINT8_N this parameter specifies the length n in bytes. For ComSignalType UINT8_DYN it specifies the maximum length in bytes. For all other types this parameter shall be ignored. Range: 0..8 for normal CAN/ LIN I-PDUs, 0..254 for normal FlexRay I-PDUs, and 0..4294967295 for I-PDUs with ComIPduType TP.		
M2 Template	**M2 Description**	
TPS_SYST	The number of bits that are used to make up the opaque type.	
M2 Parameter		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize		
Mapping Rule		**Mapping Type**
Opaque data shall always be of uint8[n] and shall always be mapped to an n-bytes sized signal.		full

BSW Module	BSW Context	
Com	Com/ComConfig/ComGwMapping/ComGwSource/ComGwSourceDescription	
BSW Parameter		**BSW Type**
ComSignalType		EcucEnumerationParamDef
BSW Description		
The AUTOSAR type of the signal. Whether or not the signal is signed or unsigned can be found by examining the value of this attribute. This type could also be used to reserved appropriate storage in AUTOSAR COM.		
M2 Template	**M2 Description**	
TPS_SYST	Describes a reusable data type on the implementation level. This will typically correspond to a typedef in C-code.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignal.networkRepresentationProps.swBaseType		
Mapping Rule		**Mapping Type**
Mapping of AUTOSAR data types (defined in the software component description) to COM Signal Types. Mapping rules are described in System Template Specification.		full

BSW Module	BSW Context	
Com	Com/ComConfig/ComGwMapping/ComGwSource/ComGwSourceDescription	
BSW Parameter		**BSW Type**
ComUpdateBitPosition		EcucIntegerParamDef
BSW Description		
Bit position of update-bit inside I-PDU. If this attribute is omitted then there is no update-bit. This setting must be consistently on sender and on receiver side. Range: 0..63 for CAN and LIN 0..2031 for FlexRay		
M2 Template	**M2 Description**	

TPS_SYST	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.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping.updateIndicationBitPosition		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Com	Com/ComConfig	
BSW Parameter		**BSW Type**
ComIPdu		EcucParamConfContainerDef
BSW Description		
Contains the configuration parameters of the AUTOSAR COM module's I-PDUs.		
M2 Template	**M2 Description**	
TPS_SYST	Represents the I-PDU's handled by Com. The SignalIPdu assembled and disassembled in AUTOSAR COM consists of one or more signals.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalIPdu		
Mapping Rule		**Mapping Type**
create container for each SignalIPdu that is transmitted by the regarded ECU.		full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComIPdu	
BSW Parameter		**BSW Type**
ComIPduCallout		EcucFunctionNameDef
BSW Description		
This parameter defines the existence and the name of a callout function for the corresponding I-PDU. If this parameter is omitted no I-PDU callout shall take place for the corresponding I-PDU.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComIPdu	
BSW Parameter		**BSW Type**
ComIPduCancellationSupport		EcucBooleanParamDef
BSW Description		
Defines for I-PDUs with ComIPduType NORMAL: If the underlying IF-modul supports cancellation of transmit requests. Defines for I-PDUs with ComIPduType TP: If the underlying TP-module supports RX and TX cancellation of ongoing requests.		
M2 Template	**M2 Description**	
M2 Parameter		

Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComIPdu	
BSW Parameter		**BSW Type**
ComIPduCounter		EcucParamConfContainerDef
BSW Description		
This optional container contains the configuration parameters of PDU Counter.		
M2 Template	**M2 Description**	
TPS_SYST	A Pdu counter is included in a predefined set of PDUs and used to ensure that a sequence of PDUs is maintained.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalIPdu.pduCounter		
Mapping Rule		**Mapping Type**
If pduCounter is aggregated by ISignalIPdu then create this container		full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComIPdu/ComIPduCounter	
BSW Parameter		**BSW Type**
ComIPduCounterErrorNotification		EcucFunctionNameDef
BSW Description		
Name of Com_CbkCounterErr callback function to be called. If this parameter is omitted no I-PDU counter mismatch notification shall take place.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComIPdu/ComIPduCounter	
BSW Parameter		**BSW Type**
ComIPduCounterSize		EcucIntegerParamDef
BSW Description		
Size of PDU Counter expressed in bits		
M2 Template	**M2 Description**	
TPS_SYST	Size of PDU Counter expressed in bits.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SignalIPduCounter.pduCounterSize		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComIPdu/ComIPduCounter	
BSW Parameter		**BSW Type**
ComIPduCounterStartPosition		EcucIntegerParamDef

BSW Description		
Position of PDU counter expressed in bits from start position of data content of I-PDU (SDU). Note that PDU counter is not allowed to cross a byte border. The parameter ComIPduCounterStartPosition shall define the bit0 of the first byte like in little endian byte order.		
M2 Template	**M2 Description**	
TPS_SYST	Position of PDU counter expressed in bits. Note that PDU counter is not allowed to cross a byte border.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SignalIPduCounter.pduCounterStartPosition		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu/ComIPduCounter	
BSW Parameter		**BSW Type**
ComIPduCounterThreshold		EcucIntegerParamDef
BSW Description		
Threshold value of I-PDU counter algorithm, see ECUC_Com_00590.		
M2 Template	**M2 Description**	
TPS_SYST	Threshold value of I-PDU counter algorithm. See AUTOSAR COM Spec for more details.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SignalIPduCounter.pduCounterThreshold		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu	
BSW Parameter		**BSW Type**
ComIPduDirection		EcucEnumerationParamDef
BSW Description		
The direction defines if this I-PDU, and therefore the contributing signals and signal groups, shall be sent or received.		
M2 Template	**M2 Description**	
TPS_SYST	communication Direction of the Connector Port (input or output Port).	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreTopology::CommConnectorPort.communicationDirection		
Mapping Rule		**Mapping Type**
Find IPduTriggering of the regarded SignalIPdu. The IPduTriggering contains a reference to an IPduPort that is aggegated by the regarded ECU. If the communicationDirection of the CommConnectorPort is "in" than the IPdu is received.		full

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu	
BSW Parameter		**BSW Type**
ComIPduGroupRef		EcucReferenceDef
BSW Description		
Reference to the I-PDU groups this I-PDU belongs to.		

M2 Template	M2 Description	
TPS_SYST	Reference to a set of SignalIPdus, which are contained in the I-Pdu Group.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalIPduGroup.iSignalIPdu		
Mapping Rule		**Mapping Type**
Find IPduGroup that points to this SignalIPdu and create the reference.		full

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu	
BSW Parameter		**BSW Type**
ComIPduHandleId		EcucIntegerParamDef
BSW Description		
The numerical value used as the ID of this I-PDU. The ComIPduHandleId is required by the API calls Com_RxIndication, Com_TpRxIndication, Com_StartOfReception and Com_CopyRxData to receive I-PDUs from the PduR (ComIP-duDirection: Receive), as well as the PduId passed to an Rx-I-PDU-callout. For Tx-I-PDUs (ComIPduDirection: Send), this handle Id is used for the APIs calls Com_TxConfirmation, Com_TriggerTransmit, Com_TriggerIPDUSend, Com_CopyTxData and Com_TpTxConfirmation to transmit respectively confirm transmissions of I-PDUs, as well as the PduId passed to the Tx-I-PDU-callout configured with ComIPduCallout and/or ComIPduTriggerTransmitCallout.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu	
BSW Parameter		**BSW Type**
ComIPduReplication		EcucParamConfContainerDef
BSW Description		
This optional container contains the information needed for each I-PDU replicated.		
M2 Template	**M2 Description**	
TPS_SYST	PDU Replication is a form of redundancy where the data content of one PDU (source) is transmitted inside a set of replica PDUs.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalIPdu.pduReplication		
Mapping Rule		**Mapping Type**
If pduReplication is defined for the SignalIPdu then create this container		full

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu/ComIPduReplication	
BSW Parameter		**BSW Type**
ComIPduReplicaRef		EcucReferenceDef
BSW Description		
Reference to replicas PduR PDUs of this IPDU.		
M2 Template	**M2 Description**	
TPS_SYST	Reference to replica PDUs of this IPDU.	
M2 Parameter		

SystemTemplate::Fibex::FibexCore::CoreCommunication::SignalIPduReplication.replicaPdus	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComIPdu/ComIPduReplication	
BSW Parameter		**BSW Type**
ComIPduReplicationQuorum		EcucIntegerParamDef
BSW Description		
The number of identical I-PDUs needed for successful voting.		
M2 Template	**M2 Description**	
TPS_SYST	The number of identical I-PDUs needed for successful voting.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SignalIPduReplication.pduReplicationVoting		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComIPdu	
BSW Parameter		**BSW Type**
ComIPduSignalGroupRef		EcucReferenceDef
BSW Description		
References to all signal groups contained in this I-Pdu		
M2 Template	**M2 Description**	
TPS_SYST	An ISignalToIPduMapping describes the mapping of ISignals to SignalIPdus and defines the position of the ISignal within an SignalIPdu.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping		
Mapping Rule		**Mapping Type**
Find ISignal in the ISignalIPdu that refers to a ISignalGroup and create reference to this Group		full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComIPdu	
BSW Parameter		**BSW Type**
ComIPduSignalProcessing		EcucEnumerationParamDef
BSW Description		
For the definition of the two modes Immediate and Deferred.		
M2 Template	**M2 Description**	
TPS_SYST	Definition of the two signal processing modes Immediate and Deferred for both Tx and Rx IPdus.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::IPduPort.iPduSignalProcessing		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**
Com	Com/ComConfig/ComIPdu

BSW Parameter		BSW Type
ComIPduSignalRef		EcucReferenceDef
BSW Description		
References to all signals contained in this I-PDU.		
M2 Template	**M2 Description**	
TPS_SYST	An ISignalToIPduMapping describes the mapping of ISignals to SignalIPdus and defines the position of the ISignal within an SignalIPdu.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping		
Mapping Rule		**Mapping Type**
Find ISignal in the IPdu which refers to a SystemSignal and create reference to this Signal.		full

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu	
BSW Parameter		**BSW Type**
ComIPduTriggerTransmitCallout		EcucFunctionNameDef
BSW Description		
If there is a trigger transmit callout defined for this I-PDU this parameter contains the name of the callout function.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu	
BSW Parameter		**BSW Type**
ComIPduType		EcucEnumerationParamDef
BSW Description		
Defines if this I-PDU is a normal I-PDU that can be sent unfragmented or if this is a large I-PDU that shall be sent via the Transport Protocol of the underlying bus.		
M2 Template	**M2 Description**	
TPS_SYST	A TpConnection represents an internal path for the transmission or reception of a Pdu via Tp and describes the sender of a connection. It can be derived from the TpConnection of the respective busses	
M2 Parameter		
SystemTemplate::TransportProtocols::TpConfig		
Mapping Rule		**Mapping Type**
If this IPdu is mapped in the System Description by a TpConnection to NPdus than set this EnumerationLiteral to TP.		full

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu	
BSW Parameter		**BSW Type**
ComPduIdRef		EcucReferenceDef
BSW Description		
Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.		

M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComIPdu	
BSW Parameter		**BSW Type**
ComTxIPdu		EcucParamConfContainerDef
BSW Description		
This container contains additional transmission related configuration parameters of the AUTOSAR COM module's I-PDUs.		
M2 Template	**M2 Description**	
TPS_SYST	Represents the IPdus handled by Com. The IPdu 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.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalIPdu		
Mapping Rule		**Mapping Type**
create container if an ISignalIPdu is transmitted by the regarded ECU.		full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComIPdu/ComTxIPdu	
BSW Parameter		**BSW Type**
ComMinimumDelayTime		EcucFloatParamDef
BSW Description		
Defines the Minimum Delay Time (MDT) between successive transmissions of this I-PDU in seconds. The MDT is independent of the possible different transmission modes. There is only one minimum delay time parameter for one I-PDU. The minimum delay timer is not reset by changing the transmission mode. Hence, it is not allowed to violate the minimum delay time by transmission mode changes. It is not possible to monitor the minimum delay time for I-PDUs that are requested using the Com_TriggerTransmit API.		
M2 Template	**M2 Description**	
TPS_SYST	Minimum Delay in seconds between successive transmissions of this I-PDU, independent of the Transmission Mode.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::IPduTiming.minimumDelay		
Mapping Rule		**Mapping Type**
Find IPduTiming for the transmitted IPdu and use the specified value.		full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComIPdu/ComTxIPdu	
BSW Parameter		**BSW Type**
ComTxIPduClearUpdateBit		EcucEnumerationParamDef
BSW Description		
Defines when the update-bits of signals or signal groups, contained in this I-PDU, will be cleared.		
M2 Template	**M2 Description**	
M2 Parameter		

Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComIPdu/ComTxIPdu	
BSW Parameter		**BSW Type**
ComTxIPduUnusedAreasDefault		EcucIntegerParamDef
BSW Description		
The AUTOSAR COM module fills not used areas of an I-PDU with this byte pattern. This attribute is mandatory to avoid undefined behaviour. This byte-pattern will be repeated throughout the I-PDU before any init-values or update-bits were set.		
M2 Template	**M2 Description**	
TPS_SYST	AUTOSAR COM fills 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.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalIPdu.unusedBitPattern		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComIPdu/ComTxIPdu	
BSW Parameter		**BSW Type**
ComTxModeFalse		EcucParamConfContainerDef
BSW Description		
This container contains the configuration parameters of the AUTOSAR COM module's transmission modes in the case the ComFilter evaluates to false.		
M2 Template	**M2 Description**	
TPS_SYST	If the COM Transmission Mode is false the timing is aggregated by the TransmissionModeTiming element (role transmissionModeFalseTiming).	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing::TransmissionModeDeclaration.transmissionModeFalseTiming		
Mapping Rule		**Mapping Type**
Create Container if a timing specification is defined for this IPdu.		full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeFalse	
BSW Parameter		**BSW Type**
ComTxMode		EcucParamConfContainerDef
BSW Description		
This container contains the configuration parameters of the AUTOSAR COM module's transmission modes.		
M2 Template	**M2 Description**	
TPS_SYST	AUTOSAR COM provides the possibility to define two different TRANSMISSION MODES for each I-PDU. The Transmission Mode of an IPdu that is valid at a specific point in time is selected using the values of the signals that are mapped to this IPdu.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing::TransmissionModeTiming		

Mapping Rule	Mapping Type
Create Container if a timing specification is defined for this IPdu.	full

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeFalse/ComTxMode	
BSW Parameter		**BSW Type**
ComTxModeMode		EcucEnumerationParamDef
BSW Description		
The available transmission modes described in [18] shall be extended by the additional mode None. The transmission mode None shall not have any further sub-attributes in the ComTxMode object.		
M2 Template	**M2 Description**	
TPS_SYST	If the COM Transmission Mode is false the timing is aggregated by the TransmitransmissionModeFalseTiming. If the COM Transmission Mode is true the timing is aggregated by the TransmissionModeTrueTiming.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing::TransmissionModeTiming		
Mapping Rule		**Mapping Type**
Periodic Mode is described by CyclicTiming. Direct /n-times Mode is described by EventControlledTiming. Mixed Mode is described if Cyclic and EventControlledTimings are assigned. None is described if no timing is assigned.		full

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeFalse/ComTxMode	
BSW Parameter		**BSW Type**
ComTxModeNumberOfRepetitions		EcucIntegerParamDef
BSW Description		
Defines the number of repetitions for the transmission mode DIRECT and the event driven part of transmission mode MIXED.		
M2 Template	**M2 Description**	
TPS_SYST	Defines the number of repetitions for the Direct/N-Times transmission mode and the event driven part of Mixed transmission mode.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing::EventControlledTiming.numberOfRepetitions		
Mapping Rule		**Mapping Type**
If "EventControlledTiming.numberOfRepetitions" = 0 then ComTxModeNumberOfRepetitions = 0; If "EventControlledTiming.numberOfRepetitions" $>$ 0 then ComTxModeNumberOfRepetitions = "EventControlledTiming.numberOfRepetitions" + 1		full

BSW Module	BSW Context	
Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeFalse/ComTxMode	
BSW Parameter		**BSW Type**
ComTxModeRepetitionPeriod		EcucFloatParamDef
BSW Description		
Defines the repetition period in seconds of the multiple transmissions in case ComTxModeNumberOfRepetitions is configured greater than 1 and ComTxModeMode is configured to DIRECT or MIXED. In case of the mixed transmission mode only the event driven part is affected.		
M2 Template	**M2 Description**	

TPS_SYST	Specification of the time in seconds that elapses before the pdu can be sent the next time (Minimum repeat gap between two pdus)
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing::EventControlledTiming.repetitionPeriod	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeFalse/ComTxMode
BSW Parameter	**BSW Type**
ComTxModeTimeOffset	EcucFloatParamDef
BSW Description	
Defines the period in seconds between the start of the I-PDU by Com_IpduGroupControl and the first transmission request in case ComTxModeMode is configured to PERIODIC or MIXED. In case of the mixed transmission mode only the periodic part is affected. In case ComTxModeTimeOffset is omitted or configured to 0, the first periodic transmission shall be transmitted within the next invocation of Com_MainFunctionTx.	
M2 Template	**M2 Description**
TPS_SYST	Specification of the time that is needed before the pdu can be sent the first time.
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing::CyclicTiming.timeOffset	
Mapping Rule	**Mapping Type**
The value for the True and the False Transmission Mode can be derived from I PduTiming.TransmissionModeDeclaration.TransmissionModeTiming element	full

BSW Module	**BSW Context**
Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeFalse/ComTxMode
BSW Parameter	**BSW Type**
ComTxModeTimePeriod	EcucFloatParamDef
BSW Description	
Defines the repetition period in seconds of the periodic transmission requests in case ComTxModeMode is configured to PERIODIC or MIXED. In case of the mixed transmission mode only the periodic part is affected.	
M2 Template	**M2 Description**
TPS_SYST	Period of the repetition of cyclic transmissions.
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing::CyclicTiming.timePeriod	
Mapping Rule	**Mapping Type**
The value for the True and the False Transmission Mode can be derived from I PduTiming.TransmissionModeDeclaration.TransmissionModeTiming element	full

BSW Module	**BSW Context**
Com	Com/ComConfig/ComIPdu/ComTxIPdu
BSW Parameter	**BSW Type**
ComTxModeTrue	EcucParamConfContainerDef
BSW Description	
This container contains the configuration parameters of the AUTOSAR COM module's transmission modes in the case the ComFilter evaluates to true.	
M2 Template	**M2 Description**

TPS_SYST	If the COM Transmission Mode is true the timing can be aggregated by the TransmissionModeTiming element (role transmissionModeTrueTiming)	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing::TransmissionModeDeclaration.transmissionModeTrueTiming		
Mapping Rule		**Mapping Type**
Create Container if a timing specification is defined for this IPdu.		full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeTrue	
BSW Parameter		**BSW Type**
ComTxMode		EcucParamConfContainerDef
BSW Description		
This container contains the configuration parameters of the AUTOSAR COM module's transmission modes.		
M2 Template	**M2 Description**	
TPS_SYST	AUTOSAR COM provides the possibility to define two different TRANSMISSION MODES for each I-PDU. The Transmission Mode of an IPdu that is valid at a specific point in time is selected using the values of the signals that are mapped to this IPdu.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing::TransmissionModeTiming		
Mapping Rule		**Mapping Type**
Create Container if a timing specification is defined for this IPdu.		full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeTrue/ComTxMode	
BSW Parameter		**BSW Type**
ComTxModeMode		EcucEnumerationParamDef
BSW Description		
The available transmission modes described in [18] shall be extended by the additional mode None. The transmission mode None shall not have any further sub-attributes in the ComTxMode object.		
M2 Template	**M2 Description**	
TPS_SYST	If the COM Transmission Mode is false the timing is aggregated by the TransmitransmissionModeFalseTiming. If the COM Transmission Mode is true the timing is aggregated by the TransmissionModeTrueTiming.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing::TransmissionModeTiming		
Mapping Rule		**Mapping Type**
Periodic Mode is described by CyclicTiming. Direct /n-times Mode is described by EventControlledTiming. Mixed Mode is described if Cyclic and EventControlledTimings are assigned. None is described if no timing is assigned.		full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeTrue/ComTxMode	
BSW Parameter		**BSW Type**
ComTxModeNumberOfRepetitions		EcucIntegerParamDef
BSW Description		

Defines the number of repetitions for the transmission mode DIRECT and the event driven part of transmission mode MIXED.		
M2 Template	**M2 Description**	
TPS_SYST	Defines the number of repetitions for the Direct/N-Times transmission mode and the event driven part of Mixed transmission mode.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing::EventControlledTiming.numberOfRepetitions		
Mapping Rule		**Mapping Type**
If "EventControlledTiming.numberOfRepetitions" = 0 then ComTxModeNumberOfRepetitions = 0; If "EventControlledTiming.numberOfRepetitions" > 0 then ComTxModeNumberOfRepetitions = "EventControlledTiming.numberOfRepetitions" + 1		full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeTrue/ComTxMode	
BSW Parameter		**BSW Type**
ComTxModeRepetitionPeriod		EcucFloatParamDef
BSW Description		
Defines the repetition period in seconds of the multiple transmissions in case ComTxModeNumberOfRepetitions is configured greater than 1 and ComTxModeMode is configured to DIRECT or MIXED. In case of the mixed transmission mode only the event driven part is affected.		
M2 Template	**M2 Description**	
TPS_SYST	Specification of the time in seconds that elapses before the pdu can be sent the next time (Minimum repeat gap between two pdus)	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing::EventControlledTiming.repetitionPeriod		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeTrue/ComTxMode	
BSW Parameter		**BSW Type**
ComTxModeTimeOffset		EcucFloatParamDef
BSW Description		
Defines the period in seconds between the start of the I-PDU by Com_IpduGroupControl and the first transmission request in case ComTxModeMode is configured to PERIODIC or MIXED. In case of the mixed transmission mode only the periodic part is affected. In case ComTxModeTimeOffset is omitted or configured to 0, the first periodic transmission shall be transmitted within the next invocation of Com_MainFunctionTx.		
M2 Template	**M2 Description**	
TPS_SYST	Specification of the time that is needed before the pdu can be sent the first time.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing::CyclicTiming.timeOffset		
Mapping Rule		**Mapping Type**
The value for the True and the False Transmission Mode can be derived from IPduTiming.TransmissionModeDeclaration.TransmissionModeTiming element		full

BSW Module	**BSW Context**

Com	Com/ComConfig/ComIPdu/ComTxIPdu/ComTxModeTrue/ComTxMode
BSW Parameter	**BSW Type**
ComTxModeTimePeriod	EcucFloatParamDef
BSW Description	
Defines the repetition period in seconds of the periodic transmission requests in case ComTxMode-Mode is configured to PERIODIC or MIXED. In case of the mixed transmission mode only the periodic part is affected.	
M2 Template	**M2 Description**
TPS_SYST	Period of the repetition of cyclic transmissions.
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::Timing::CyclicTiming.timePeriod	
Mapping Rule	**Mapping Type**
The value for the True and the False Transmission Mode can be derived from I PduTiming.TransmissionModeDeclaration.TransmissionModeTiming element	full

BSW Module	**BSW Context**
Com	Com/ComConfig
BSW Parameter	**BSW Type**
ComIPduGroup	EcucParamConfContainerDef
BSW Description	
Contains the configuration parameters of the AUTOSAR COM module's I-PDU groups.	
M2 Template	**M2 Description**
TPS_SYST	The AUTOSAR COM Layer is able to start and to stop sending and receiving configurable groups of I-Pdus during runtime. An I-Pdu group contains either Com I-Pdus or I-Pdu groups.
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalIPduGroup	
Mapping Rule	**Mapping Type**
Create container for each CoreCommunication::ISignalIPduGroup that is contained in the ECU Extract.	full

BSW Module	**BSW Context**
Com	Com/ComConfig/ComIPduGroup
BSW Parameter	**BSW Type**
ComIPduGroupGroupRef	EcucReferenceDef
BSW Description	
References to all I-PDU groups that includes this I-PDU group. If this reference is omitted this I-PDU group does not belong to another I-PDU group.	
M2 Template	**M2 Description**
TPS_SYST	An I-PDU group can be included in other I-Pdu groups.
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalIPduGroup.containedISignalIPduGroup	
Mapping Rule	**Mapping Type**
If the IPduGroup has a reference to a contained IPduGroup then create this reference.	full

BSW Module	**BSW Context**
Com	Com/ComConfig/ComIPduGroup
BSW Parameter	**BSW Type**
ComIPduGroupHandleId	EcucIntegerParamDef

BSW Description		
The numerical value used as the ID of this I-PDU Group . The ComIPduGroupHandleId is required by the API calls to start and stop I-PDU Groups. Range: 0 .. (ComSupportedIPduGroups-1)		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
Com	Com/ComConfig	
BSW Parameter		**BSW Type**
ComMaxIPduCnt		EcucIntegerParamDef
BSW Description		
Maximum number of IPdus. This parameter is needed only in case of post-build loadable implementation using static memory allocation.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
Com	Com/ComConfig	
BSW Parameter		**BSW Type**
ComSignal		EcucParamConfContainerDef
BSW Description		
Contains the configuration parameters of the AUTOSAR COM module's signals.		
M2 Template	**M2 Description**	
TPS_SYST	An ISignalToIPduMapping describes the mapping of ISignals to SignalIPdus and defines the position of the ISignal within an SignalIPdu.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping		
Mapping Rule		**Mapping Type**
Tx: If an ISignal has no ISignalPort assigned a ComSignal shall always be created in the transmitting ECUs in order to send the init value. Rx: If an ISignal has no ISignalPort assigned there is no need for the existence of a ComSignal in receiving ECU		full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComSignal	
BSW Parameter		**BSW Type**
ComBitPosition		EcucIntegerParamDef
BSW Description		
Starting position within the I-PDU. This parameter refers to the position in the I-PDU and not in the shadow buffer. If the endianness conversion is configured to Opaque the parameter ComBitPosition shall define the bit0 of the first byte like in little endian byte order		

M2 Template	M2 Description
TPS_SYST	This parameter is necessary to describe the bitposition of a signal within an SignalIPdu.
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping.startPosition	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	BSW Context
Com	Com/ComConfig/ComSignal
BSW Parameter	**BSW Type**
ComBitSize	EcucIntegerParamDef
BSW Description	
Size in bits, for integer signal types. For ComSignalType UINT8_N and UINT8_DYN the size shall be configured by ComSignalLength. For ComSignalTypes FLOAT32 and FLOAT64 the size is already defined by the signal type and therefore may be omitted.	
M2 Template	**M2 Description**
TPS_SYST	Size of the signal in bits.
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignal.length	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	BSW Context
Com	Com/ComConfig/ComSignal
BSW Parameter	**BSW Type**
ComDataInvalidAction	EcucEnumerationParamDef
BSW Description	
This parameter defines the action performed upon reception of an invalid signal. Relating to signal groups the action in case if one of the included signals is an invalid signal. If Replace is used the ComSignalInitValue will be used for the replacement.	
M2 Template	**M2 Description**
TPS_SWCT	InvalidationPolicy for a particular dataElement
M2 Parameter	
SWComponentTemplate::PortInterface::InvalidationPolicy	
Mapping Rule	**Mapping Type**
If strategy HandleInvalidEnum.keep is defined then set parameter to notify. If strategy HandleInvalidEnum.replace is defined then set parameter to replace. If the parameter does not exist this corresponds to the value HandleInvalid Enum.dontInvalidate.	full

BSW Module	BSW Context
Com	Com/ComConfig/ComSignal
BSW Parameter	**BSW Type**
ComErrorNotification	EcucFunctionNameDef
BSW Description	
Only valid on sender side: Name of Com_CbkTxErr callback function to be called. If this parameter is omitted no error notification shall take place.	
M2 Template	**M2 Description**
M2 Parameter	

Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComSignal	
BSW Parameter		**BSW Type**
ComFilter		EcucParamConfContainerDef
BSW Description		
This container contains the configuration parameters of the AUTOSAR COM module's Filters. Note: On sender side the container is used to specify the transmission mode conditions.		
M2 Template	**M2 Description**	
TPS_SWCT, TPS_SYST	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.	
M2 Parameter		
CommonStructure::Filter::DataFilter		
Mapping Rule		**Mapping Type**
Create container on the receiver side if the NonqueuedReceiverComSpec contains a DataFilter. Create Container on the sender side if the TransmissionMode Condition element contains a reference to this signal.		full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComSignal/ComFilter	
BSW Parameter		**BSW Type**
ComFilterAlgorithm		EcucEnumerationParamDef
BSW Description		
The range of values is specified in the [17] specification, chapter 2.2.2, Reception Filtering.		
M2 Template	**M2 Description**	
TPS_SWCT, TPS_SYST	This attriburte specifies the type of the filter.	
M2 Parameter		
CommonStructure::Filter::DataFilter.dataFilterType		
Mapping Rule		**Mapping Type**
Mapping between DataFilterTypeEnum and ComFilterAlgorithm Enum is necessary.		full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComSignal/ComFilter	
BSW Parameter		**BSW Type**
ComFilterMask		EcucIntegerParamDef
BSW Description		
The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering.		
M2 Template	**M2 Description**	
TPS_SWCT, TPS_SYST	Mask for old and new value.	
M2 Parameter		
CommonStructure::Filter::DataFilter.mask		
Mapping Rule		**Mapping Type**

1:1 mapping	full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComSignal/ComFilter	
BSW Parameter		**BSW Type**
ComFilterMax		EcucIntegerParamDef
BSW Description		
The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering.		
M2 Template	**M2 Description**	
TPS_SWCT, TPS_SYST	Value to specify the upper boundary	
M2 Parameter		
CommonStructure::Filter::DataFilter.max		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComSignal/ComFilter	
BSW Parameter		**BSW Type**
ComFilterMin		EcucIntegerParamDef
BSW Description		
The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering.		
M2 Template	**M2 Description**	
TPS_SWCT, TPS_SYST	Value to specify the lower boundary	
M2 Parameter		
CommonStructure::Filter::DataFilter.min		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComSignal/ComFilter	
BSW Parameter		**BSW Type**
ComFilterOffset		EcucIntegerParamDef
BSW Description		
The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering. Range = 0..(ComFilterPeriod-1)		
M2 Template	**M2 Description**	
TPS_SWCT, TPS_SYST	Specifies the initial number of messages to occur before the first message is passed	
M2 Parameter		
CommonStructure::Filter::DataFilter.offset		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**

Com	Com/ComConfig/ComSignal/ComFilter		
BSW Parameter		**BSW Type**	
ComFilterPeriod		EcucIntegerParamDef	
BSW Description			
This parameter defines the period of the ComFilterAlgorithm ONE_EVERY_N.			
M2 Template	**M2 Description**		
TPS_SWCT, TPS_SYST	specifies number of messages to occur before the message is passed again		
M2 Parameter			
CommonStructure::Filter::DataFilter.period			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	**BSW Context**		
Com	Com/ComConfig/ComSignal/ComFilter		
BSW Parameter		**BSW Type**	
ComFilterX		EcucIntegerParamDef	
BSW Description			
The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering.			
M2 Template	**M2 Description**		
TPS_SWCT, TPS_SYST	Value to compare with		
M2 Parameter			
CommonStructure::Filter::DataFilter.x			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	**BSW Context**		
Com	Com/ComConfig/ComSignal		
BSW Parameter		**BSW Type**	
ComFirstTimeout		EcucFloatParamDef	
BSW Description			
Defines the length of the first deadline monitoring timeout period in seconds. This timeout is used immediately after start (or restart) of the deadline monitoring service. The timeout period of the successive periods is configured by ECUC_Com_00263.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComSignal	
BSW Parameter		**BSW Type**
ComHandleId		EcucIntegerParamDef
BSW Description		

The numerical value used as the ID. For signals it is required by the API calls Com_UpdateShadowSignal, Com_ReceiveShadowSignal and Com_InvalidateShadowSignal. For signals groups it is required by the Com_SendSignalGroup and Com_ReceiveSignalGroup calls.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComSignal	
BSW Parameter		**BSW Type**
ComInitialValueOnly		EcucBooleanParamDef
BSW Description		
This parameter defines that the respective signal's initial value shall be put into the respective PDU but there will not be any update of the value through the RTE. Thus the Com implementation does not need to expect any API calls for this signal (group).		
M2 Template	**M2 Description**	
TPS_SYST	Whether an ECU actually participates in the communication (Tx or Rx) is defined via the association of the TriggeringElements to Port Elements of the CommunicationConnector.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalPort		
Mapping Rule		**Mapping Type**
Tx: If an ISignal has no ISignalPort assigned a ComSignal shall always be created in the transmitting ECUs in order to send the init value. Rx: If an ISignal has no ISignalPort assigned there is no need for the existence of a ComSignal in the rec. Ecu		full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComSignal	
BSW Parameter		**BSW Type**
ComInvalidNotification		EcucFunctionNameDef
BSW Description		
Only valid on receiver side: Name of Com_CbkInv callback function to be called. Name of the function which notifies the RTE about the reception of an invalidated signal/ signal group. Only applicable if ComDataInvalidAction is configured to NOTIFY.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComSignal	
BSW Parameter		**BSW Type**

ComNotification		EcucFunctionNameDef	
BSW Description			
On sender side: Name of Com_CbkTxAck callback function to be called. On receiver side: Name of Com_CbkRxAck callback function to be called. If this parameter is omitted no notification shall take place.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
Com	Com/ComConfig/ComSignal		
BSW Parameter		**BSW Type**	
ComRxDataTimeoutAction		EcucEnumerationParamDef	
BSW Description			
This parameter defines the action performed upon expiration of the reception deadline monitoring timer.			
M2 Template	**M2 Description**		
TPS_SYST	Strategies of handling a reception timeout violation.		
M2 Parameter			
SWComponentTemplate::Communication::NonqueuedReceiverComSpec.handleTimeoutType			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	**BSW Context**		
Com	Com/ComConfig/ComSignal		
BSW Parameter		**BSW Type**	
ComSignalDataInvalidValue		EcucStringParamDef	
BSW Description			
Defines the data invalid value of the signal. In case the ComSignalType is UINT8, UINT16, UINT32, SINT8, SINT16, SINT32 the string shall be interpreted as defined in the chapter Integer Type in the AUTOSAR EcuC specification. In case the ComSignalType is FLOAT32, FLOAT64 the string shall be interpreted as defined in the chapter Float Type in the AUTOSAR EcuC specification. In case the ComSignalType is BOOLEAN the string shall be interpreted as defined in the chapter Boolean Type in the AUTOSAR EcuC specification. In case the ComSignal is a UINT8_N, UINT8_DYN the string shall be interpreted as a decimal representation of the characters separated by blanks, e.g. "97 98 100" means a string "abd", where the char "a" is in byte 0(lowest address), "b" is in byte 1, and "d" is in byte 2 and (highest address). For the ComSignalType UINT8_DYN the dynamic length shall be set to the number of configured characters. An empty string "" shall be interpretated as 0-sized dynamic signal.			
M2 Template	**M2 Description**		
TPS_SWCT, TPS_SYST	Optional value to express invalidity of the actual data element.		
M2 Parameter			
DataDictionary::DataDefProperties::SwDataDefProps.invalidValue			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignal	
BSW Parameter		**BSW Type**
ComSignalEndianness		EcucEnumerationParamDef
BSW Description		
Defines the endianness of the signal's network representation.		
M2 Template	**M2 Description**	
TPS_SYST	This parameter defines the order of the bytes of the signal and the packing into the IPdu. The byte ordering Little Endian (MostSignificantByteLast), Big Endian (MostSignificantByteFirst) and Opaque can be selected.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping.packingByteOrder		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignal	
BSW Parameter		**BSW Type**
ComSignalInitValue		EcucStringParamDef
BSW Description		
Initial value for this signal. In case of UINT8_N the default value is a string of length ComSignalLength with all bytes set to 0x00. In case of UINT8_DYN the initial size shall be 0. In case the ComSignalType is UINT8, UINT16, UINT32, SINT8, SINT16, SINT32 the string shall be interpreted as defined in the chapter Integer Type in the AUTOSAR EcuC specification. In case the ComSignalType is FLOAT32, FLOAT64 the string shall be interpreted as defined in the chapter Float Type in the AUTOSAR EcuC specification. In case the ComSignalType is BOOLEAN the string shall be interpreted as defined in the chapter Boolean Type in the AUTOSAR EcuC specification. In case the ComSignal is a UINT8_N, UINT8_DYN the string shall be interpreted as a decimal representation of the characters separated by blanks, e.g. "97 98 100" means a string "abd", where the char "a" is in byte 0(lowest address), "b" is in byte 1, and "d" is in byte 2 and (highest address). For the ComSignalType UINT8_DYN the dynamic length shall be set to the number of configured characters. An empty string "" shall be interpretated as 0-sized dynamic signal.		
M2 Template	**M2 Description**	
TPS_SWCT, TPS_SYST	If a full DataMapping exist this information may be available from a configured SenderComSpec and ReceiverComSpec. In case the System Description doesn't use a complete Software ComponentDescription an optional reference from SystemSignal is used.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignal.initValue,SWComponentTemplate::Communication::NonqueuedSenderComSpec.initValue		
Mapping Rule		**Mapping Type**
It is possible to aggregate an initValue at the level of a ComSpec in the SW C Template. in case the System Description doesn't use a complete Software Component Description (VFB View) the initValue is defined in the System Template.		full

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignal	
BSW Parameter		**BSW Type**

ComSignalLength		EcucIntegerParamDef	
BSW Description			
Description: For ComSignalType UINT8_N this parameter specifies the length n in bytes. For ComSignalType UINT8_DYN it specifies the maximum length in bytes. For all other types this parameter shall be ignored. Range: 0..8 for normal CAN/ LIN I-PDUs, 0..254 for normal FlexRay I-PDUs, and 0..4294967295 for I-PDUs with ComIPduType TP.			
M2 Template	**M2 Description**		
TPS_SYST	The number of bits that are used to make up the opaque type.		
M2 Parameter			
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize			
Mapping Rule			**Mapping Type**
Opaque data shall always be of uint8[n] and shall always be mapped to an n-bytes sized signal.			full

BSW Module	**BSW Context**		
Com	Com/ComConfig/ComSignal		
BSW Parameter		**BSW Type**	
ComSignalType		EcucEnumerationParamDef	
BSW Description			
The AUTOSAR type of the signal. Whether or not the signal is signed or unsigned can be found by examining the value of this attribute. This type could also be used to reserved appropriate storage in AUTOSAR COM.			
M2 Template	**M2 Description**		
TPS_SYST	Describes a reusable data type on the implementation level. This will typically correspond to a typedef in C-code.		
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignal.networkRepresentationProps.swBaseType			
Mapping Rule			**Mapping Type**
Mapping of AUTOSAR data types (defined in the software component description) to COM Signal Types. Mapping rules are described in System Template Specification.			full

BSW Module	**BSW Context**		
Com	Com/ComConfig/ComSignal		
BSW Parameter		**BSW Type**	
ComSystemTemplateSystemSignalRef		EcucForeignReferenceDef	
BSW Description			
Reference to the ISignalToIPduMapping that contains a reference to the ISignal (System Template) which this ComSignal (or ComGroupSignal) represents.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	**BSW Context**

Com	Com/ComConfig/ComSignal		
BSW Parameter		**BSW Type**	
ComTimeout		EcucFloatParamDef	
BSW Description			
Defines the length of the deadline monitoring timeout period in seconds. The period for the first timeout period can be configured separately by ECUC_Com_00183.			
M2 Template	**M2 Description**		
TPS_SWCT, TPS_SYST	Timeout value in seconds for the reception of the ISignal.		
M2 Parameter			
SWComponentTemplate::Communication::NonqueuedReceiverComSpec.aliveTimeout,SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalPort.timeout			
Mapping Rule			**Mapping Type**
If a full DataMapping exist for the SystemSignal this information may be available from a configured NonqueuedReceiverComSpec. In this case the timeout value in ReceiverComSpec overrides the optional timeout specification in the System Template.			full

BSW Module	**BSW Context**		
Com	Com/ComConfig/ComSignal		
BSW Parameter		**BSW Type**	
ComTimeoutNotification		EcucFunctionNameDef	
BSW Description			
On sender side: Name of Com_CbkTxTOut callback function to be called. On receiver side: Name of Com_CbkRxTOut callback function to be called.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
Com	Com/ComConfig/ComSignal		
BSW Parameter		**BSW Type**	
ComTransferProperty		EcucEnumerationParamDef	
BSW Description			
Defines if a write access to this signal can trigger the transmission of the corresponding I-PDU. If the I-PDU is triggered, depends also on the transmission mode of the corresponding I-PDU.			
M2 Template	**M2 Description**		
TPS_SYST	The triggered transfer property causes immediate transmission of the IPdu, except if transmission mode Periodic or transmission mode NONE is defined for the IPdu. The Pending transfer property does not cause transmission of an I-PDU.		
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping.transferProperty			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	**BSW Context**
Com	Com/ComConfig/ComSignal

BSW Parameter		BSW Type
ComUpdateBitPosition		EcucIntegerParamDef
BSW Description		
Bit position of update-bit inside I-PDU. If this attribute is omitted then there is no update-bit. This setting must be consistently on sender and on receiver side. Range: 0..63 for CAN and LIN 0..2031 for FlexRay		
M2 Template	**M2 Description**	
TPS_SYST	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.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping.updateIndication BitPosition		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Com	Com/ComConfig	
BSW Parameter		**BSW Type**
ComSignalGroup		EcucParamConfContainerDef
BSW Description		
Contains the configuration parameters of the AUTOSAR COM module's signal groups.		
M2 Template	**M2 Description**	
TPS_SYST	An ISignalGroup refers to a set of ISignals that must always be kept together. A ISignalGroup represents a COM Signal Group.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalGroup		
Mapping Rule		**Mapping Type**
Create this container for each ISignalGroup that exist in the ECU Extract.		full

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup	
BSW Parameter		**BSW Type**
ComDataInvalidAction		EcucEnumerationParamDef
BSW Description		
This parameter defines the action performed upon reception of an invalid signal. Relating to signal groups the action in case if one of the included signals is an invalid signal. If Replace is used the ComSignalInitValue will be used for the replacement.		
M2 Template	**M2 Description**	
TPS_SWCT	InvalidationPolicy for a particular dataElement	
M2 Parameter		
SWComponentTemplate::PortInterface::InvalidationPolicy		
Mapping Rule		**Mapping Type**
If strategy HandleInvalidEnum.keep is defined then set parameter to notify. If strategy HandleInvalidEnum.replace is defined then set parameter to replace. If the parameter does not exist this corresponds to the value HandleInvalid Enum.dontInvalidate.		full

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup	
BSW Parameter		**BSW Type**
ComErrorNotification		EcucFunctionNameDef
BSW Description		
Only valid on sender side: Name of Com_CbkTxErr callback function to be called. If this parameter is omitted no error notification shall take place.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup	
BSW Parameter		**BSW Type**
ComFirstTimeout		EcucFloatParamDef
BSW Description		
Defines the length of the first deadline monitoring timeout period in seconds. This timeout is used immediately after start (or restart) of the deadline monitoring service. The timeout period of the successive periods is configured by ECUC_Com_00263.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup	
BSW Parameter		**BSW Type**
ComGroupSignal		EcucParamConfContainerDef
BSW Description		
This container contains the configuration parameters of group signals. I.e. signals that are included within a signal group.		
M2 Template	**M2 Description**	
TPS_SYST	Signal of the Interaction Layer.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignal		
Mapping Rule		**Mapping Type**
Create Container for each ISignal that is contained in the ISignalGroup.		full

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal	
BSW Parameter		**BSW Type**
ComBitPosition		EcucIntegerParamDef
BSW Description		
Starting position within the I-PDU. This parameter refers to the position in the I-PDU and not in the shadow buffer. If the endianness conversion is configured to Opaque the parameter ComBitPosition shall define the bit0 of the first byte like in little endian byte order		

M2 Template	M2 Description	
TPS_SYST	This parameter is necessary to describe the bitposition of a signal within an SignalIPdu.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping.startPosition		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal	
BSW Parameter		**BSW Type**
ComBitSize		EcucIntegerParamDef
BSW Description		
Size in bits, for integer signal types. For ComSignalType UINT8_N and UINT8_DYN the size shall be configured by ComSignalLength. For ComSignalTypes FLOAT32 and FLOAT64 the size is already defined by the signal type and therefore may be omitted.		
M2 Template	**M2 Description**	
TPS_SYST	Size of the signal in bits.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignal.length		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal	
BSW Parameter		**BSW Type**
ComFilter		EcucParamConfContainerDef
BSW Description		
This container contains the configuration parameters of the AUTOSAR COM module's Filters. Note: On sender side the container is used to specify the transmission mode conditions.		
M2 Template	**M2 Description**	
TPS_SWCT, TPS_SYST	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.	
M2 Parameter		
CommonStructure::Filter::DataFilter		
Mapping Rule		**Mapping Type**
Create container on the receiver side if the NonqueuedReceiverComSpec contains a DataFilter. Create Container on the sender side if the TransmissionMode Condition element contains a reference to this signal.		full

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal/ComFilter	
BSW Parameter		**BSW Type**
ComFilterAlgorithm		EcucEnumerationParamDef
BSW Description		
The range of values is specified in the [17] specification, chapter 2.2.2, Reception Filtering.		
M2 Template	**M2 Description**	
TPS_SWCT, TPS_SYST	This attriburte specifies the type of the filter.	

M2 Parameter	
CommonStructure::Filter::DataFilter.dataFilterType	
Mapping Rule	**Mapping Type**
Mapping between DataFilterTypeEnum and ComFilterAlgorithm Enum is necessary.	full

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal/ComFilter	
BSW Parameter		**BSW Type**
ComFilterMask		EcucIntegerParamDef
BSW Description		
The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering.		
M2 Template	**M2 Description**	
TPS_SWCT, TPS_SYST	Mask for old and new value.	
M2 Parameter		
CommonStructure::Filter::DataFilter.mask		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal/ComFilter	
BSW Parameter		**BSW Type**
ComFilterMax		EcucIntegerParamDef
BSW Description		
The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering.		
M2 Template	**M2 Description**	
TPS_SWCT, TPS_SYST	Value to specify the upper boundary	
M2 Parameter		
CommonStructure::Filter::DataFilter.max		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal/ComFilter	
BSW Parameter		**BSW Type**
ComFilterMin		EcucIntegerParamDef
BSW Description		
The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering.		
M2 Template	**M2 Description**	
TPS_SWCT, TPS_SYST	Value to specify the lower boundary	
M2 Parameter		
CommonStructure::Filter::DataFilter.min		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal/ComFilter	
BSW Parameter		**BSW Type**
ComFilterOffset		EcucIntegerParamDef
BSW Description		
The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering. Range = 0..(ComFilterPeriod-1)		
M2 Template	**M2 Description**	
TPS_SWCT, TPS_SYST	Specifies the initial number of messages to occur before the first message is passed	
M2 Parameter		
CommonStructure::Filter::DataFilter.offset		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal/ComFilter	
BSW Parameter		**BSW Type**
ComFilterPeriod		EcucIntegerParamDef
BSW Description		
This parameter defines the period of the ComFilterAlgorithm ONE_EVERY_N.		
M2 Template	**M2 Description**	
TPS_SWCT, TPS_SYST	specifies number of messages to occur before the message is passed again	
M2 Parameter		
CommonStructure::Filter::DataFilter.period		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal/ComFilter	
BSW Parameter		**BSW Type**
ComFilterX		EcucIntegerParamDef
BSW Description		
The name of this attribute corresponds to the parameter name in the [17] specification of Reception Filtering.		
M2 Template	**M2 Description**	
TPS_SWCT, TPS_SYST	Value to compare with	
M2 Parameter		
CommonStructure::Filter::DataFilter.x		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal	
BSW Parameter		**BSW Type**
ComHandleId		EcucIntegerParamDef

BSW Description	
The numerical value used as the ID. For signals it is required by the API calls Com_UpdateShadowSignal, Com_ReceiveShadowSignal and Com_InvalidateShadowSignal. For signals groups it is required by the Com_SendSignalGroup and Com_ReceiveSignalGroup calls.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal	
BSW Parameter		**BSW Type**
ComSignalDataInvalidValue		EcucStringParamDef
BSW Description		
Defines the data invalid value of the signal. In case the ComSignalType is UINT8, UINT16, UINT32, SINT8, SINT16, SINT32 the string shall be interpreted as defined in the chapter Integer Type in the AUTOSAR EcuC specification. In case the ComSignalType is FLOAT32, FLOAT64 the string shall be interpreted as defined in the chapter Float Type in the AUTOSAR EcuC specification. In case the ComSignalType is BOOLEAN the string shall be interpreted as defined in the chapter Boolean Type in the AUTOSAR EcuC specification. In case the ComSignal is a UINT8_N, UINT8_DYN the string shall be interpreted as a decimal representation of the characters separated by blanks, e.g. "97 98 100" means a string "abd", where the char "a" is in byte 0(lowest address), "b" is in byte 1, and "d" is in byte 2 and (highest address). For the ComSignalType UINT8_DYN the dynamic length shall be set to the number of configured characters. An empty string "" shall be interpretated as 0-sized dynamic signal.		
M2 Template	**M2 Description**	
TPS_SWCT, TPS_SYST	Optional value to express invalidity of the actual data element.	
M2 Parameter		
DataDictionary::DataDefProperties::SwDataDefProps.invalidValue		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal	
BSW Parameter		**BSW Type**
ComSignalEndianness		EcucEnumerationParamDef
BSW Description		
Defines the endianness of the signal's network representation.		
M2 Template	**M2 Description**	
TPS_SYST	This parameter defines the order of the bytes of the signal and the packing into the IPdu. The byte ordering Little Endian (MostSignificantByteLast), Big Endian (MostSignificantByteFirst) and Opaque can be selected.	
M2 Parameter		

SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping.packingByte Order	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal	
BSW Parameter		**BSW Type**
ComSignalInitValue		EcucStringParamDef
BSW Description		
Initial value for this signal. In case of UINT8_N the default value is a string of length ComSignal-Length with all bytes set to 0x00. In case of UINT8_DYN the initial size shall be 0. In case the ComSignalType is UINT8, UINT16, UINT32, SINT8, SINT16, SINT32 the string shall be interpreted as defined in the chapter Integer Type in the AUTOSAR EcuC specification. In case the ComSignalType is FLOAT32, FLOAT64 the string shall be interpreted as defined in the chapter Float Type in the AUTOSAR EcuC specification. In case the ComSignalType is BOOLEAN the string shall be interpreted as defined in the chapter Boolean Type in the AUTOSAR EcuC specification. In case the ComSignal is a UINT8_N, UINT8_DYN the string shall be interpreted as a decimal representation of the characters separated by blanks, e.g. "97 98 100" means a string "abd", where the char "a" is in byte 0(lowest address), "b" is in byte 1, and "d" is in byte 2 and (highest address). For the ComSignalType UINT8_DYN the dynamic length shall be set to the number of configured characters. An empty string "" shall be interpretated as 0-sized dynamic signal.		
M2 Template	**M2 Description**	
TPS_SWCT, TPS_SYST	If a full DataMapping exist this information may be available from a configured SenderComSpec and ReceiverComSpec. In case the System Description doesn't use a complete Software ComponentDescription an optional reference from SystemSignal is used.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignal.initValue,SWComponentTemplate::Communication::NonqueuedSenderComSpec.initValue		
Mapping Rule		**Mapping Type**
It is possible to aggregate an initValue at the level of a ComSpec in the SW C Template. in case the System Description doesn't use a complete Software Component Description (VFB View) the initValue is defined in the System Template.		full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal	
BSW Parameter		**BSW Type**
ComSignalLength		EcucIntegerParamDef
BSW Description		
Description: For ComSignalType UINT8_N this parameter specifies the length n in bytes. For ComSignalType UINT8_DYN it specifies the maximum length in bytes. For all other types this parameter shall be ignored. Range: 0..8 for normal CAN/ LIN I-PDUs, 0..254 for normal FlexRay I-PDUs, and 0..4294967295 for I-PDUs with ComIPduType TP.		
M2 Template	**M2 Description**	
TPS_SYST	The number of bits that are used to make up the opaque type.	

M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
Mapping Rule	**Mapping Type**
Opaque data shall always be of uint8[n] and shall always be mapped to an n-bytes sized signal.	full

BSW Module	BSW Context		
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal		
BSW Parameter		**BSW Type**	
ComSignalType		EcucEnumerationParamDef	
BSW Description			
The AUTOSAR type of the signal. Whether or not the signal is signed or unsigned can be found by examining the value of this attribute. This type could also be used to reserved appropriate storage in AUTOSAR COM.			
M2 Template	**M2 Description**		
TPS_SYST	Describes a reusable data type on the implementation level. This will typically correspond to a typedef in C-code.		
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignal.networkRepresentationProps.swBaseType			
Mapping Rule			**Mapping Type**
Mapping of AUTOSAR data types (defined in the software component description) to COM Signal Types. Mapping rules are described in System Template Specification.			full

BSW Module	BSW Context		
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal		
BSW Parameter		**BSW Type**	
ComSystemTemplateSystemSignalRef		EcucForeignReferenceDef	
BSW Description			
Reference to the ISignalToIPduMapping that contains a reference to the ISignal (System Template) which this ComSignal (or ComGroupSignal) represents.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	BSW Context		
Com	Com/ComConfig/ComSignalGroup/ComGroupSignal		
BSW Parameter		**BSW Type**	
ComTransferProperty		EcucEnumerationParamDef	
BSW Description			
Optionally defines whether this group signal shall contribute to the TRIGGERED_ON_CHANGE transfer property of the signal group. If at least one group signal of a signal group has the "ComTransferProperty" configured all other group signals of that signal group shall have the attribute configured as well.			
M2 Template	**M2 Description**		

TPS_SYST	The triggered transfer property causes immediate transmission of the IPdu, except if transmission mode Periodic or transmission mode NONE is defined for the IPdu. The Pending transfer property does not cause transmission of an I-PDU.
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping.transferProperty	
Mapping Rule	**Mapping Type**
ISignalToIPduMapping element contains a reference to the ISignalGroup and contains the attribute "transferProperty"	full

BSW Module	**BSW Context**
Com	Com/ComConfig/ComSignalGroup
BSW Parameter	**BSW Type**
ComHandleId	EcucIntegerParamDef
BSW Description	
The numerical value used as the ID. For signals it is required by the API calls Com_UpdateShadowSignal, Com_ReceiveShadowSignal and Com_InvalidateShadowSignal. For signals groups it is required by the Com_SendSignalGroup and Com_ReceiveSignalGroup calls.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
Com	Com/ComConfig/ComSignalGroup
BSW Parameter	**BSW Type**
ComInitialValueOnly	EcucBooleanParamDef
BSW Description	
This parameter defines that the respective signal's initial value shall be put into the respective PDU but there will not be any update of the value through the RTE. Thus the Com implementation does not need to expect any API calls for this signal (group).	
M2 Template	**M2 Description**
TPS_SYST	Whether an ECU actually participates in the communication (Tx or Rx) is defined via the association of the TriggeringElements to Port Elements of the CommunicationConnector.
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalPort	
Mapping Rule	**Mapping Type**
Tx: If an ISignal has no ISignalPort assigned a ComSignal shall always be created in the transmitting ECUs in order to send the init value. Rx: If an ISignal has no ISignalPort assigned there is no need for the existence of a ComSignal in the rec. Ecu	full

BSW Module	**BSW Context**
Com	Com/ComConfig/ComSignalGroup
BSW Parameter	**BSW Type**

ComInvalidNotification	EcucFunctionNameDef
BSW Description	
Only valid on receiver side: Name of Com_CbkInv callback function to be called. Name of the function which notifies the RTE about the reception of an invalidated signal/ signal group. Only applicable if ComDataInvalidAction is configured to NOTIFY.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
Com	Com/ComConfig/ComSignalGroup
BSW Parameter	**BSW Type**
ComNotification	EcucFunctionNameDef
BSW Description	
On sender side: Name of Com_CbkTxAck callback function to be called. On receiver side: Name of Com_CbkRxAck callback function to be called. If this parameter is omitted no notification shall take place.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
Com	Com/ComConfig/ComSignalGroup
BSW Parameter	**BSW Type**
ComRxDataTimeoutAction	EcucEnumerationParamDef
BSW Description	
This parameter defines the action performed upon expiration of the reception deadline monitoring timer.	
M2 Template	**M2 Description**
TPS_SYST	Strategies of handling a reception timeout violation.
M2 Parameter	
SWComponentTemplate::Communication::NonqueuedReceiverComSpec.handleTimeoutType	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
Com	Com/ComConfig/ComSignalGroup
BSW Parameter	**BSW Type**
ComSystemTemplateSignalGroupRef	EcucForeignReferenceDef
BSW Description	
Reference to the ISignalToIPduMapping that contains a reference to the ISignalGroup (SystemTemplate) which this ComSignalGroup represents.	
M2 Template	**M2 Description**

M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComSignalGroup	
BSW Parameter		**BSW Type**
ComTimeout		EcucFloatParamDef
BSW Description		
Defines the length of the deadline monitoring timeout period in seconds. The period for the first timeout period can be configured separately by ECUC_Com_00183.		
M2 Template	**M2 Description**	
TPS_SWCT, TPS_SYST	Timeout value in seconds for the reception of the ISignal.	
M2 Parameter		
SWComponentTemplate::Communication::NonqueuedReceiverComSpec.aliveTimeout,SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalPort.timeout		
Mapping Rule		**Mapping Type**
If a full DataMapping exist for the SystemSignal this information may be available from a configured NonqueuedReceiverComSpec. In this case the timeout value in ReceiverComSpec overrides the optional timeout specification in the System Template.		full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComSignalGroup	
BSW Parameter		**BSW Type**
ComTimeoutNotification		EcucFunctionNameDef
BSW Description		
On sender side: Name of Com_CbkTxTOut callback function to be called. On receiver side: Name of Com_CbkRxTOut callback function to be called.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComSignalGroup	
BSW Parameter		**BSW Type**
ComTransferProperty		EcucEnumerationParamDef
BSW Description		
Defines if a write access to this signal can trigger the transmission of the corresponding I-PDU. If the I-PDU is triggered, depends also on the transmission mode of the corresponding I-PDU.		
M2 Template	**M2 Description**	
TPS_SYST	The triggered transfer property causes immediate transmission of the IPdu, except if transmission mode Periodic or transmission mode NONE is defined for the IPdu. The Pending transfer property does not cause transmission of an I-PDU.	

M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping.transferProperty	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	BSW Context	
Com	Com/ComConfig/ComSignalGroup	
BSW Parameter		**BSW Type**
ComUpdateBitPosition		EcucIntegerParamDef
BSW Description		
Bit position of update-bit inside I-PDU. If this attribute is omitted then there is no update-bit. This setting must be consistently on sender and on receiver side. Range: 0..63 for CAN and LIN 0..2031 for FlexRay		
M2 Template	**M2 Description**	
TPS_SYST	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.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping.updateIndication BitPosition		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Com	Com/ComConfig	
BSW Parameter		**BSW Type**
ComTimeBase		EcucParamConfContainerDef
BSW Description		
Contains the timebase parameters for Tx, Rx and routing.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Com	Com/ComConfig/ComTimeBase	
BSW Parameter		**BSW Type**
ComGwTimeBase		EcucFloatParamDef
BSW Description		

The period between successive calls to Com_MainFunctionRouteSignals in seconds. This parameter may be used by the COM generator to transform the values of the signal gateway related timing configuration parameters of the COM module to internal implementation specific counter or tick values. The COM module's internal timing handling is implementation specific. The COM module (generator) might rely on the fact that Com_MainFunctionRouteSignals is scheduled according to the value configured here.		
M2 Template	**M2 Description**	
TPS_SYST	The period between successive calls to Com_MainFunctionRouteSignals of the AUTOSAR COM module in seconds.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreTopology::EcuInstance.comConfigurationGwTimeBase		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**		
Com	Com/ComConfig/ComTimeBase		
BSW Parameter		**BSW Type**	
ComRxTimeBase		EcucFloatParamDef	
BSW Description			
The period between successive calls to Com_MainFunctionRx in seconds. This parameter may be used by the COM generator to transform the values of the reception related timing configuration parameters of the COM module to internal implementation specific counter or tick values. The COM module's internal timing handling is implementation specific. The COM module (generator) may rely on the fact that Com_MainFunctionRx is scheduled according to the value configured here.			
M2 Template	**M2 Description**		
TPS_SYST	The period between successive calls to Com_MainFunctionRx of the AUTOSAR COM module in seconds.		
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreTopology::EcuInstance.comConfigurationRxTimeBase			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	**BSW Context**	
Com	Com/ComConfig/ComTimeBase	
BSW Parameter		**BSW Type**
ComTxTimeBase		EcucFloatParamDef
BSW Description		
The period between successive calls to Com_MainFunctionTx in seconds. This parameter may be used by the COM generator to transform the values of the transmission related timing configuration parameters of the COM module to internal implementation specific counter or tick values. The COM module's internal timing handling is implementation specific. The COM module (generator) may rely on the fact that Com_MainFunctionTx is scheduled according to the value configured here.		
M2 Template	**M2 Description**	
TPS_SYST	The period between successive calls to Com_MainFunctionTx of the AUTOSAR COM module in seconds.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreTopology::EcuInstance.comConfigurationTxTimeBase		

Mapping Rule	Mapping Type
1:1 mapping	full

BSW Module	BSW Context	
Com	Com	
BSW Parameter		**BSW Type**
ComGeneral		EcucParamConfContainerDef
BSW Description		
Contains the general configuration parameters of the module.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
Com	Com/ComGeneral	
BSW Parameter		**BSW Type**
ComCancellationSupport		EcucBooleanParamDef
BSW Description		
This parameter enables/disables the cancellation feature: true: enabled false: disabled		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
Com	Com/ComGeneral	
BSW Parameter		**BSW Type**
ComConfigurationUseDet		EcucBooleanParamDef
BSW Description		
The error hook shall contain code to call the Det. If this parameter is configured COM_DEV_ERROR_DETECT shall be set to ON as output of the configuration tool. (as input for the source code).		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Com	Com/ComGeneral	
BSW Parameter		**BSW Type**

ComEnableMDTForCyclicTransmission	EcucBooleanParamDef	
BSW Description		
Enables globally for the whole Com module the minimum delay time monitoring for cyclic and repeated transmissions (ComTxModeMode=PERIODIC or ComTxModeMode=MIXED for the cyclic transmissions, ComTxModeNumberOfRepetitions $>$ 0 for repeated transmissions).		
M2 Template	**M2 Description**	
TPS_SYST	Enables for the Com module of this EcuInstance the minimum delay time monitoring for cyclic and repeated transmissions (TransmissionModeTiming has cyclicTiming assigned or eventControlledTiming with numberOfRepetitions $>$ 0)	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreTopology::EcuInstance.comEnableMDTForCyclicTransmission		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Com	Com/ComGeneral	
BSW Parameter		**BSW Type**
ComRetryFailedTransmitRequests		EcucBooleanParamDef
BSW Description		
If this Parameter is set to true, retry of failed transmission requests is enabled. If this Parameter is not present, the default value is assumed.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
Com	Com/ComGeneral	
BSW Parameter		**BSW Type**
ComSupportedIPduGroups		EcucIntegerParamDef
BSW Description		
Defines the maximum number of supported I-PDU groups.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
Com	Com/ComGeneral	
BSW Parameter		**BSW Type**
ComUserCbkHeaderFile		EcucStringParamDef
BSW Description		
Defines the header files for callback functions which shall be included by the COM module.		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context	
Com	Com/ComGeneral	
BSW Parameter		**BSW Type**
ComVersionInfoApi		EcucBooleanParamDef
BSW Description		
Activate/Deactivate the version information API (Com_GetVersionInfo). True: version information API activated False: version information API deactivated		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

D.1.2 IPduM Mapping

BSW Module	BSW Context	
IpduM	IpduM	
BSW Parameter		**BSW Type**
IpduMConfig		EcucParamConfContainerDef
BSW Description		
This container contains the sub containers of the IpduM module. The IpduMTxPathway subcontainer includes information about sent I-PDUs. The IpduMRxPathway includes information about received I-PDUs. This container is a MultipleConfigurationContainer, i.e. this container and its sub-containers exist once per configuration set.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig	
BSW Parameter		**BSW Type**
IpduMMaxTxBufferSize		EcucIntegerParamDef
BSW Description		
Maximum total size of all Tx buffers. This parameter is needed only in case of post-build loadable implementation using static memory allocation.		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig	
BSW Parameter		**BSW Type**
IpduMMaxTxPathwayCnt		EcucIntegerParamDef
BSW Description		
Maximum number of transmitted IPdus. This parameter is needed only in case of post-build loadable implementation using static memory allocation.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig	
BSW Parameter		**BSW Type**
IpduMRxPathway		EcucParamConfContainerDef
BSW Description		
Contains the configuration parameters received I-PDUs by the IpduM module.		
M2 Template	**M2 Description**	
TPS_SYST	A MultiplexedPdu (i.e. NOT a COM I-PDU) contains a DynamicPart, an optional StaticPart and a selectorField. In case of multiplexing this IPdu is routed between the Pdu Multiplexer and the Interface Layer.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::MultiplexedIPdu		
Mapping Rule		**Mapping Type**
Create container for each received multiplexed Ipdu (IPduTriggering that references the MultiplexedIPdu contains a reference to an "In" Pdu Port.		full

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMRxPathway	
BSW Parameter		**BSW Type**
IpduMRxIndication		EcucParamConfContainerDef
BSW Description		
Contains the configuration for incoming RxIndication calls.		
M2 Template	**M2 Description**	
TPS_SYST	A MultiplexedPdu (i.e. NOT a COM I-PDU) contains a DynamicPart, an optional StaticPart and a selectorField. In case of multiplexing this IPdu is routed between the Pdu Multiplexer and the Interface Layer.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::MultiplexedIPdu		
Mapping Rule		**Mapping Type**
Create container for each received multiplexed Ipdu (IPduTriggering that references the MultiplexedIPdu contains a reference to an "In" Pdu Port		full

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication	
BSW Parameter		**BSW Type**
IpduMByteOrder		EcucEnumerationParamDef
BSW Description		
This parameter defines the ByteOrder for all segments (static and dynamic part) and for the selectorField within the MultiplexedPdu. The absolute position of a segment in the MultiplexedIPdu is determined by the definition of the ByteOrder parameter: If BIG_ENDIAN is specified, the SegmentPosition indicates the bit position of the most significant bit in an IPDU. If LITTLE_ENDIAN is specified, the SegmentPosition indicates the bit position of the least significant bit in an IPDU.		
M2 Template	**M2 Description**	
TPS_SYST	This attribute defines the order of the bytes of the segment and the packing into the MultiplexedIPdu.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::MultiplexedIPdu.selectorFieldByteOrder, SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition.segmentByteOrder		
Mapping Rule		**Mapping Type**
A mix between Little Endian and Big Endian within a MultiplexedIPdu is not allowed.		full

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication	
BSW Parameter		**BSW Type**
IpduMRxDynamicPart		EcucParamConfContainerDef
BSW Description		
This container contains the configuration for the dynamic part of incoming RxIndication calls. When an incoming received I-PDU's selector field matches the IpduMRxSelectorValue, the new outgoing I-PDU for the dynamic part is constructed as defined by the segments (defined in the IpduMDynamicSegment container) and sent out with the I-PDU ID referenced by IpduMOutgoingDynamicPduRef. In case no dynamic part shall be extracted from this received I-PDU this container does not exist. This use-case can occur in case a MultiplexedIPdu is received by an ECU which is only interested in the static part of the MultiplexedIPdu.		
M2 Template	**M2 Description**	
TPS_SYST	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.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::DynamicPartAlternative		
Mapping Rule		**Mapping Type**
Create container for each DynamicPartAlternative of the MultiplexedIPdu.		full

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication/IpduMRxDynamic Part	
BSW Parameter		**BSW Type**
IpduMOutgoingDynamicPduRef		EcucReferenceDef
BSW Description		

When the new I-PDU is sent out it is sent with this I-PDU ID. Reference to the sent PDU representation in the ECU Configuration Description exchange file.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication/IpduMRxDynamicPart	
BSW Parameter		**BSW Type**
IpduMRxSelectorValue		EcucIntegerParamDef
BSW Description		
This is the selector value that this container refers to.		
M2 Template	**M2 Description**	
TPS_SYST	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.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::DynamicPartAlternative.selectorFieldCode		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication/IpduMRxDynamicPart	
BSW Parameter		**BSW Type**
IpduMSegment		EcucParamConfContainerDef
BSW Description		
Please note that this container is deprecated and will be removed in the future. Old description: This contains the location and the length of a segment. A segment must fit inside the I-PDU. The segment in the source I-PDU that is located at the IpduMSegmentPosition is copied to the same position in the destination I-PDU.		
M2 Template	**M2 Description**	
TPS_SYST	The StaticPart and the DynamicPart can be separated in multiple segments within the multiplexed PDU.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition		
Mapping Rule		**Mapping Type**
Source bit fields and the destination bit position can be derived from the segmentPosition.		full

BSW Module	**BSW Context**	
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication/IpduMRxDynamicPart/IpduMSegment	
BSW Parameter		**BSW Type**

IpduMSegmentLength		EcucIntegerParamDef	
BSW Description			
Length of the segment in bits.			
M2 Template	**M2 Description**		
TPS_SYST	With the attributes startPositionMultiplexedIPdu and startPositionInSignalIPdu a segment in the source can be copied to the segment in the destination.		
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition.segmentLength			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	**BSW Context**		
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication/IpduMRxDynamicPart/IpduMSegment		
BSW Parameter		**BSW Type**	
IpduMSegmentPosition		EcucIntegerParamDef	
BSW Description			
Segments bit position in the multiplexed Pdu.			
M2 Template	**M2 Description**		
TPS_SYST	With the attributes startPositionMultiplexedIPdu and startPositionInSignalIPdu a segment in the source can be copied to the segment in the destination.		
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition.segmentPosition			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	**BSW Context**		
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication		
BSW Parameter		**BSW Type**	
IpduMRxDynamicSegment		EcucParamConfContainerDef	
BSW Description			
The dynamic part of the multiplexed incoming I-Pdu (referenced by IpduMRxIndicationPduRef) can be separated into several segments. For each segment one IpduMRxDynamicSegment container shall be created that contains the location and the length of the segment. Please note that each configured segment will be copied into the destination I-Pdu that is referenced in the IpduMRxDynamicPart container and will be copied from the same location in the multiplexed incoming I-Pdu. The segment layout for all dynamic Parts is always identical.			
M2 Template	**M2 Description**		
TPS_SYST	The StaticPart and the DynamicPart can be separated in multiple segments within the multiplexed PDU.		
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition			
Mapping Rule			**Mapping Type**
Shall be derived from segmentPosition elements that are aggregated by the DynamicPart.			full

BSW Module	**BSW Context**
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication/IpduMRxDynamicSegment

BSW Parameter			BSW Type
IpduMSegmentLength			EcucIntegerParamDef
BSW Description			
Length of the segment in bits.			
M2 Template	**M2 Description**		
TPS_SYST	With the attributes startPositionMultiplexedIPdu and startPositionInSignalIPdu a segment in the source can be copied to the segment in the destination.		
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition.segmentLength			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	BSW Context		
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication/IpduMRxDynamicSegment		
BSW Parameter		**BSW Type**	
IpduMSegmentPosition		EcucIntegerParamDef	
BSW Description			
Segments bit position in the multiplexed Pdu.			
M2 Template	**M2 Description**		
TPS_SYST	With the attributes startPositionMultiplexedIPdu and startPositionInSignalIPdu a segment in the source can be copied to the segment in the destination.		
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition.segmentPosition			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	BSW Context		
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication		
BSW Parameter		**BSW Type**	
IpduMRxHandleId		EcucIntegerParamDef	
BSW Description			
This is the I-PDU ID of the incoming I-PDU. If an incoming RxIndication's I-PDU ID matches this value then it is unpacked according to the specification in this container.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication	
BSW Parameter		**BSW Type**
IpduMRxIndicationPduRef		EcucReferenceDef
BSW Description		
Reference to the received Pdu representation in the ECU Configuration Description exchange file.		
M2 Template	**M2 Description**	
M2 Parameter		

Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**	
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication	
BSW Parameter		**BSW Type**
IpduMRxStaticPart		EcucParamConfContainerDef
BSW Description		
This container contains the configuration for the static part of incoming RxIndication calls. On reception, the new outgoing I-PDU for the static part is constructed as defined by the segments (defined in the IpduMStaticSegment container) and sent out with the I-PDU ID referenced by IpduMOutgoingStaticPduRef.		
M2 Template	**M2 Description**	
TPS_SYST	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.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::StaticPart		
Mapping Rule		**Mapping Type**
Create container if StaticPart exists in the MultiplexedIPdu.		full

BSW Module	**BSW Context**	
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication/IpduMRxStaticPart	
BSW Parameter		**BSW Type**
IpduMOutgoingStaticPduRef		EcucReferenceDef
BSW Description		
When the new I-PDU is sent out it is sent with this I-PDU ID. Reference to the sent Pdu representation in the ECU Configuration Description exchange file.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication/IpduMRxStaticPart	
BSW Parameter		**BSW Type**
IpduMSegment		EcucParamConfContainerDef
BSW Description		
Please note that this container is deprecated and will be removed in the future. Old description: This contains the location and the length of a segment. A segment must fit inside the I-PDU. The segment in the source I-PDU that is located at the IpduMSegmentPosition is copied to the same position in the destination I-PDU.		
M2 Template	**M2 Description**	
TPS_SYST	The StaticPart and the DynamicPart can be separated in multiple segments within the multiplexed PDU.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition		

Mapping Rule	Mapping Type
Source bit fields and the destination bit position can be derived from the segmentPosition.	full

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication/IpduMRxStaticPart/IpduMSegment	
BSW Parameter		**BSW Type**
IpduMSegmentLength		EcucIntegerParamDef
BSW Description		
Length of the segment in bits.		
M2 Template	**M2 Description**	
TPS_SYST	With the attributes startPositionMultiplexedIPdu and startPositionInSignalIPdu a segment in the source can be copied to the segment in the destination.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition.segmentLength		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication/IpduMRxStaticPart/IpduMSegment	
BSW Parameter		**BSW Type**
IpduMSegmentPosition		EcucIntegerParamDef
BSW Description		
Segments bit position in the multiplexed Pdu.		
M2 Template	**M2 Description**	
TPS_SYST	With the attributes startPositionMultiplexedIPdu and startPositionInSignalIPdu a segment in the source can be copied to the segment in the destination.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition.segmentPosition		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication	
BSW Parameter		**BSW Type**
IpduMRxStaticSegment		EcucParamConfContainerDef
BSW Description		
The static part of the multiplexed incoming I-Pdu (referenced by IpduMRxIndicationPduRef) can be separated into several segments. For each segment one IpduMRxStaticSegment container shall be created that contains the location and the length of the segment. Please note that each configured segment will be copied into the destination I-Pdu that is referenced in the IpduMRxStaticPart container and will be copied from the same location in the multiplexed incoming I-Pdu.		
M2 Template	**M2 Description**	
TPS_SYST	The StaticPart and the DynamicPart can be separated in multiple segments within the multiplexed PDU.	
M2 Parameter		

SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition	
Mapping Rule	**Mapping Type**
Shall be derived from segmentPosition elements that are aggregated by the StaticPart.	full

BSW Module	**BSW Context**	
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication/IpduMRxStaticSegment	
BSW Parameter		**BSW Type**
IpduMSegmentLength		EcucIntegerParamDef
BSW Description		
Length of the segment in bits.		
M2 Template	**M2 Description**	
TPS_SYST	With the attributes startPositionMultiplexedIPdu and startPositionInSignalIPdu a segment in the source can be copied to the segment in the destination.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition.segmentLength		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication/IpduMRxStaticSegment	
BSW Parameter		**BSW Type**
IpduMSegmentPosition		EcucIntegerParamDef
BSW Description		
Segments bit position in the multiplexed Pdu.		
M2 Template	**M2 Description**	
TPS_SYST	With the attributes startPositionMultiplexedIPdu and startPositionInSignalIPdu a segment in the source can be copied to the segment in the destination.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition.segmentPosition		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication	
BSW Parameter		**BSW Type**
IpduMSelectorField		EcucParamConfContainerDef
BSW Description		
This contains the location and the length of the selector field.		
M2 Template	**M2 Description**	
TPS_SYST	The StaticPart and the DynamicPart can be separated in multiple segments within the multiplexed PDU.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition		
Mapping Rule		**Mapping Type**
Can be derived from the segmentPosition.		full

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication/IpduMSelectorField	
BSW Parameter		**BSW Type**
IpduMSelectorFieldLength		EcucIntegerParamDef
BSW Description		
Length of the selector field in bits.		
M2 Template	**M2 Description**	
TPS_SYST	With the attributes startPositionMultiplexedIPdu and startPositionInSignalIPdu a segment in the source can be copied to the segment in the destination.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::MultiplexedIPdu.selectorFieldLength		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMRxPathway/IpduMRxIndication/IpduMSelectorField	
BSW Parameter		**BSW Type**
IpduMSelectorFieldPosition		EcucIntegerParamDef
BSW Description		
Selector field bit position in the multiplexed Pdu.		
M2 Template	**M2 Description**	
TPS_SYST	With the attributes startPositionMultiplexedIPdu and startPositionInSignalIPdu a segment in the source can be copied to the segment in the destination.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::MultiplexedIPdu.selectorFieldStartPosition		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig	
BSW Parameter		**BSW Type**
IpduMTxPathway		EcucParamConfContainerDef
BSW Description		
Contains the configuration parameters transmitted I-PDUs by the IpduM module.		
M2 Template	**M2 Description**	
TPS_SYST	A MultiplexedPdu (i.e. NOT a COM I-PDU) contains a DynamicPart, an optional StaticPart and a selectorField. In case of multiplexing this IPdu is routed between the Pdu Multiplexer and the Interface Layer.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::MultiplexedIPdu		
Mapping Rule		**Mapping Type**
Create container for each transmitted multiplexed Ipdu (IPduTriggering that references the MultiplexedIPdu contains a reference to an "Out" Pdu Port.		full

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMTxPathway	
BSW Parameter		**BSW Type**
IpduMTxRequest		EcucParamConfContainerDef
BSW Description		

This container is used to specify the configuration for Transmit requests. There will be one instance of this container for each I-PDU that can be requested for transmission (the outgoing I-PDUs) by the IpduM.		
M2 Template	**M2 Description**	
TPS_SYST	A MultiplexedPdu (i.e. NOT a COM I-PDU) contains a DynamicPart, an optional StaticPart and a selectorField. In case of multiplexing this IPdu is routed between the Pdu Multiplexer and the Interface Layer.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::MultiplexedIPdu		
Mapping Rule		**Mapping Type**
Create container for each transmitted multiplexed Ipdu		full

BSW Module	**BSW Context**	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest	
BSW Parameter		**BSW Type**
IpduMByteOrder		EcucEnumerationParamDef
BSW Description		
This parameter defines the ByteOrder for all segments (static and dynamic part) and for the selectorField within the MultiplexedPdu. The absolute position of a segment in the MultiplexedIPdu is determined by the definition of the ByteOrder parameter: If BIG_ENDIAN is specified, the SegmentPosition indicates the bit position of the most significant bit in an IPDU. If LITTLE_ENDIAN is specified, the SegmentPosition indicates the bit position of the least significant bit in an IPDU.		
M2 Template	**M2 Description**	
TPS_SYST	This attribute defines the order of the bytes of the segment and the packing into the MultiplexedIPdu.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::MultiplexedIPdu.selectorFieldByteOrder, SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition.segmentByteOrder		
Mapping Rule		**Mapping Type**
A mix between Little Endian and Big Endian within a MultiplexedIPdu is not allowed.		full

BSW Module	**BSW Context**	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest	
BSW Parameter		**BSW Type**
IpduMIPduUnusedAreasDefault		EcucIntegerParamDef
BSW Description		
IpduM module fills not used areas of an I-PDU with this bit-pattern If this attribute is omitted the IpduM module does not fill the I-PDU.		
M2 Template	**M2 Description**	
TPS_SYST	AUTOSAR COM fills 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.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::MultiplexedIPdu.unusedBitPattern		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest	
BSW Parameter		**BSW Type**
IpduMInitialDynamicPart		EcucReferenceDef
BSW Description		
Reference to the dynamic part that shall be used to initialize this multiplexed TX-I-PDU.		
M2 Template	**M2 Description**	
TPS_SYST	Dynamic part that shall be used to initialize this multiplexed IPdu.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::DynamicPartAlternative.initialDynamicPart		
Mapping Rule		**Mapping Type**
If the attribute initialDynamicPart is set to true then create this reference.		full

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest	
BSW Parameter		**BSW Type**
IpduMOutgoingPduRef		EcucReferenceDef
BSW Description		
Reference to the PDU defining the outgoing I-PDU. When the outgoing I-PDU is sent this is the I-PDU ID to give it. It is the IpduM I-PDU ID of the assembled I-PDU.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest	
BSW Parameter		**BSW Type**
IpduMSelectorField		EcucParamConfContainerDef
BSW Description		
This contains the location and the length of the selector field.		
M2 Template	**M2 Description**	
TPS_SYST	The StaticPart and the DynamicPart can be separated in multiple segments within the multiplexed PDU.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition		
Mapping Rule		**Mapping Type**
Can be derived from the segmentPosition.		full

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMSelectorField	
BSW Parameter		**BSW Type**
IpduMSelectorFieldLength		EcucIntegerParamDef
BSW Description		
Length of the selector field in bits.		
M2 Template	**M2 Description**	

TPS_SYST	With the attributes startPositionMultiplexedIPdu and startPositionInSignalIPdu a segment in the source can be copied to the segment in the destination.
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::MultiplexedIPdu.selectorFieldLength	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMSelectorField
BSW Parameter	**BSW Type**
IpduMSelectorFieldPosition	EcucIntegerParamDef
BSW Description	
Selector field bit position in the multiplexed Pdu.	
M2 Template	**M2 Description**
TPS_SYST	With the attributes startPositionMultiplexedIPdu and startPositionInSignalIPdu a segment in the source can be copied to the segment in the destination.
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::MultiplexedIPdu.selectorFieldStartPosition	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest
BSW Parameter	**BSW Type**
IpduMTxConfirmationPduId	EcucIntegerParamDef
BSW Description	
Handle Id used by the PduR for confirmation (IpduM_TxConfirmation) and for TriggerTransmit (IpduM_TriggerTransmit). The existence of this parameter is essential for the PduR generation tool to actually find a symbolicNameValue for the OutgoingPdu.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest
BSW Parameter	**BSW Type**
IpduMTxConfirmationTimeout	EcucFloatParamDef
BSW Description	
This timeout (in seconds) defines the timeout period for monitoring the reception of the TxConfirmation. It is not used when an I-PDU is requested using the trigger transmit API.	
M2 Template	**M2 Description**
M2 Parameter	

Mapping Rule	Mapping Type

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest	
BSW Parameter		**BSW Type**
IpduMTxDynamicPart		EcucParamConfContainerDef
BSW Description		
Configuration parameters for an instance of a TxRequest call into the IpduM. When a Tx Request with the IpduMTxDynamicHandleId is received by the IpduM, all segments (defined in the IpduMDynamicSegment container) are copied from the incoming I-PDU into the outgoing I-PDU buffer and then the send mode honored. This container is used by the dynamic part of a TxRequest configuration. Therefore, for each outgoing I-PDU there will be one instance of this container for the dynamic part.		
M2 Template	**M2 Description**	
TPS_SYST	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.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::DynamicPartAlternative		
Mapping Rule		**Mapping Type**
Create container for each DynamicPartAlternative of the MultiplexedIPdu.		full

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxDynamicPart	
BSW Parameter		**BSW Type**
IpduMJitUpdate		EcucBooleanParamDef
BSW Description		
If configured to true fetch the data of this part Just-In-Time via the triggerTransmit API of the PduR.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxDynamicPart	
BSW Parameter		**BSW Type**
IpduMSegment		EcucParamConfContainerDef
BSW Description		
Please note that this container is deprecated and will be removed in the future. Old description: This contains the location and the length of a segment. A segment must fit inside the I-PDU. The segment in the source I-PDU that is located at the IpduMSegmentPosition is copied to the same position in the destination I-PDU.		
M2 Template	**M2 Description**	
TPS_SYST	The StaticPart and the DynamicPart can be separated in multiple segments within the multiplexed PDU.	
M2 Parameter		

SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition	
Mapping Rule	**Mapping Type**
Source bit fields and the destination bit position can be derived from the segmentPosition.	full

BSW Module	**BSW Context**	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxDynamicPart/IpduMSegment	
BSW Parameter		**BSW Type**
IpduMSegmentLength		EcucIntegerParamDef
BSW Description		
Length of the segment in bits.		
M2 Template	**M2 Description**	
TPS_SYST	With the attributes startPositionMultiplexedIPdu and startPositionInSignalIPdu a segment in the source can be copied to the segment in the destination.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition.segmentLength		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxDynamicPart/IpduMSegment	
BSW Parameter		**BSW Type**
IpduMSegmentPosition		EcucIntegerParamDef
BSW Description		
Segments bit position in the multiplexed Pdu.		
M2 Template	**M2 Description**	
TPS_SYST	With the attributes startPositionMultiplexedIPdu and startPositionInSignalIPdu a segment in the source can be copied to the segment in the destination.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition.segmentPosition		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxDynamicPart	
BSW Parameter		**BSW Type**
IpduMTxDynamicConfirmation		EcucBooleanParamDef
BSW Description		
A transmit request can be confirmed by the lower layer. If this parameter is set to true a confirmation of the I-PDU in COM representing the dynamic part is generated.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxDynamicPart	
BSW Parameter		**BSW Type**
IpduMTxDynamicHandleId		EcucIntegerParamDef
BSW Description		
This defines an incoming handle id. When the handle of an incoming Tx Request matches this id, the configured dynamic segments are copied and the IpduMTxTriggerMode is honored.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxDynamicPart	
BSW Parameter		**BSW Type**
IpduMTxDynamicPduRef		EcucReferenceDef
BSW Description		
Reference to the Pdu representation in the ECU Configuration Description exchange file to be transmitted.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest	
BSW Parameter		**BSW Type**
IpduMTxDynamicSegment		EcucParamConfContainerDef
BSW Description		
The dynamic part of the multiplexed outgoing I-Pdu (referenced by IpduMOutgoingPduRef) can be separated into several segments. For each segment one IpduMTxDynamicSegment container shall be created that contains the location and the length of the segment. Please note that each configured segment will be copied out of the source I-Pdu that is referenced in the IpduMTxDynamicPart container and will be copied to the same location in the multiplexed outgoing I-Pdu. The segment layout for all dynamic Parts is always identical.		
M2 Template	**M2 Description**	
TPS_SYST	The StaticPart and the DynamicPart can be separated in multiple segments within the multiplexed PDU.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition		
Mapping Rule		**Mapping Type**
Shall be derived from segmentPosition elements that are aggregated by the DynamicPart.		full

BSW Module	BSW Context

IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxDynamic Segment	
BSW Parameter		**BSW Type**
IpduMSegmentLength		EcucIntegerParamDef
BSW Description		
Length of the segment in bits.		
M2 Template	**M2 Description**	
TPS_SYST	With the attributes startPositionMultiplexedIPdu and startPositionInSignalIPdu a segment in the source can be copied to the segment in the destination.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition.segmentLength		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxDynamic Segment	
BSW Parameter		**BSW Type**
IpduMSegmentPosition		EcucIntegerParamDef
BSW Description		
Segments bit position in the multiplexed Pdu.		
M2 Template	**M2 Description**	
TPS_SYST	With the attributes startPositionMultiplexedIPdu and startPositionInSignalIPdu a segment in the source can be copied to the segment in the destination.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition.segmentPosition		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest	
BSW Parameter		**BSW Type**
IpduMTxStaticPart		EcucParamConfContainerDef
BSW Description		
Configuration parameters for an instance of a Tx_Request call into the IpduM. When a Tx Request with the IpduMTxStaticHandleId is received by the IpduM, all segments (defined in the IpduMStaticSegment container) are copied from the incoming I-PDU into the outgoing I-PDU buffer and then the send mode honored. This container is used for the static part of a TxRequest configuration. Therefore, for each outgoing I-PDU there will be one instance of this container for the static part if it exists.		
M2 Template	**M2 Description**	
TPS_SYST	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.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::StaticPart		
Mapping Rule		**Mapping Type**
Create container if StaticPart exists in the MultiplexedIPdu.		full

BSW Module	**BSW Context**	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxStaticPart	
BSW Parameter		**BSW Type**

IpduMJitUpdate		EcucBooleanParamDef	
BSW Description			
If configured to true fetch the data of this part Just-In-Time via the triggerTransmit API of the PduR.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxStaticPart		
BSW Parameter		**BSW Type**	
IpduMSegment		EcucParamConfContainerDef	
BSW Description			
Please note that this container is deprecated and will be removed in the future. Old description: This contains the location and the length of a segment. A segment must fit inside the I-PDU. The segment in the source I-PDU that is located at the IpduMSegmentPosition is copied to the same position in the destination I-PDU.			
M2 Template	**M2 Description**		
TPS_SYST	The StaticPart and the DynamicPart can be separated in multiple segments within the multiplexed PDU.		
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition			
Mapping Rule			**Mapping Type**
Source bit fields and the destination bit position can be derived from the segmentPosition.			full

BSW Module	**BSW Context**		
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxStaticPart/IpduMSegment		
BSW Parameter		**BSW Type**	
IpduMSegmentLength		EcucIntegerParamDef	
BSW Description			
Length of the segment in bits.			
M2 Template	**M2 Description**		
TPS_SYST	With the attributes startPositionMultiplexedIPdu and startPositionInSignalIPdu a segment in the source can be copied to the segment in the destination.		
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition.segmentLength			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	**BSW Context**	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxStaticPart/IpduMSegment	
BSW Parameter		**BSW Type**
IpduMSegmentPosition		EcucIntegerParamDef
BSW Description		

Segments bit position in the multiplexed Pdu.		
M2 Template	**M2 Description**	
TPS_SYST	With the attributes startPositionMultiplexedIPdu and startPositionInSignalIPdu a segment in the source can be copied to the segment in the destination.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition.segmentPosition		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxStaticPart	
BSW Parameter		**BSW Type**
IpduMTxStaticConfirmation		EcucBooleanParamDef
BSW Description		
A transmit request can be confirmed by the lower layer. If this parameter is set to true a confirmation of the I-PDU in COM representing the static part is generated.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxStaticPart	
BSW Parameter		**BSW Type**
IpduMTxStaticHandleId		EcucIntegerParamDef
BSW Description		
This defines an incoming handle id. When the handle of an incoming Tx Request matches this id, the configured static segments are copied and the IpduMTxTriggerMode is honored.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxStaticPart	
BSW Parameter		**BSW Type**
IpduMTxStaticPduRef		EcucReferenceDef
BSW Description		
Reference to the Pdu representation in the ECU Configuration Description exchange file to be transmitted.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context		
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest		
BSW Parameter		**BSW Type**	
IpduMTxStaticSegment		EcucParamConfContainerDef	
BSW Description			
The static part of the multiplexed outgoing I-Pdu (referenced by IpduMOutgoingPduRef) can be separated into several segments. For each segment one IpduMTxStaticSegment container shall be created that contains the location and the length of the segment. Please note that each segment in the source I-Pdu that is referenced in the IpduMTxStatic-Part container will be copied to the same location in the multiplexed outgoing I-Pdu.			
M2 Template	**M2 Description**		
TPS_SYST	The StaticPart and the DynamicPart can be separated in multiple segments within the multiplexed PDU.		
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition			
Mapping Rule			**Mapping Type**
Shall be derived from segmentPosition elements that are aggregated by the StaticPart.			full

BSW Module	BSW Context		
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxStaticSeg-ment		
BSW Parameter		**BSW Type**	
IpduMSegmentLength		EcucIntegerParamDef	
BSW Description			
Length of the segment in bits.			
M2 Template	**M2 Description**		
TPS_SYST	With the attributes startPositionMultiplexedIPdu and startPositionInSignalIPdu a segment in the source can be copied to the segment in the destination.		
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition.segmentLength			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	BSW Context		
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest/IpduMTxStaticSeg-ment		
BSW Parameter		**BSW Type**	
IpduMSegmentPosition		EcucIntegerParamDef	
BSW Description			
Segments bit position in the multiplexed Pdu.			
M2 Template	**M2 Description**		
TPS_SYST	With the attributes startPositionMultiplexedIPdu and startPositionInSignalIPdu a segment in the source can be copied to the segment in the destination.		
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreCommunication::SegmentPosition.segmentPosition			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	BSW Context	
IpduM	IpduM/IpduMConfig/IpduMTxPathway/IpduMTxRequest	
BSW Parameter		**BSW Type**
IpduMTxTriggerMode		EcucEnumerationParamDef
BSW Description		
Selects whether to send the multiplexed I-PDU immediately or at some later date.		
M2 Template	**M2 Description**	
TPS_SYST	IPduM can be configured to send a transmission request for the new multiplexed I-PDU to the PDU-Router because of the trigger conditions/ modes that are described in the TriggerMode enumeration.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::MultiplexedIPdu.triggerMode		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
IpduM	IpduM	
BSW Parameter		**BSW Type**
IpduMGeneral		EcucParamConfContainerDef
BSW Description		
Contains the general configuration parameters of IpduM.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
IpduM	IpduM/IpduMGeneral	
BSW Parameter		**BSW Type**
IpduMConfigurationTimeBase		EcucFloatParamDef
BSW Description		
The cycle time with which IpduM_MainFunction should be invoked (in seconds).		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
IpduM	IpduM/IpduMGeneral	
BSW Parameter		**BSW Type**
IpduMDevErrorDetect		EcucBooleanParamDef
BSW Description		
Active/Deactivate the detection of development errors, for production code this parameter has to be False. True: error detection activated False: error detection deactivated		

M2 Template	M2 Description	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context		
IpduM	IpduM/IpduMGeneral		
BSW Parameter		**BSW Type**	
IpduMStaticPartExists		EcucBooleanParamDef	
BSW Description			
This is to allow optimizations in the case the IpduM will never be used with a static part. Note that this is a pre-compile option. If this is set to False then it will not be possible to add static parts after compilation. True: A static part may exist. False: A static part will never exist.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
IpduM	IpduM/IpduMGeneral		
BSW Parameter		**BSW Type**	
IpduMVersionInfoApi		EcucBooleanParamDef	
BSW Description			
Active/Deactivate the version information API. true: version information activated false: version information deactivated			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context	
IpduM	IpduM	
BSW Parameter		**BSW Type**
IpduMPublishedInformation		EcucParamConfContainerDef
BSW Description		
Additional published parameters not covered by CommonPublishedInformation container. Note that these parameters do not have any configuration class setting, since they are published information.		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	BSW Context	
IpduM	IpduM/IpduMPublishedInformation	
BSW Parameter		**BSW Type**
IpduMRxDirectComInvocation		EcucBooleanParamDef
BSW Description		
If set to TRUE the COM invocation optimization as defined in IPDUM140 is implemented.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

D.1.3 PduR

BSW Module	BSW Context	
PduR	PduR	
BSW Parameter		**BSW Type**
PduRBswModules		EcucParamConfContainerDef
BSW Description		
Each container describes a specific BSW module (upper/CDD/lower/IpduM) that the PDU Router shall interface to. The reason to have it as own configuration container instead of implication of the routing path is to be able to configure CDD:s properly and to force module's to be used in a post-build situation even though no routing is made to/from this module (future configurations may include these modules).		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
PduR	PduR/PduRBswModules	
BSW Parameter		**BSW Type**
PduRBswModuleRef		EcucForeignReferenceDef
BSW Description		
This is a reference to one BSW module's configuration (i.e. not the ECUC parameter definition template). Example, there could be several configurations of LinIf and this reference selects one of them.		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
PduR	PduR/PduRBswModules
BSW Parameter	**BSW Type**
PduRCancelReceive	EcucBooleanParamDef
BSW Description	
Specifies if the Transport protocol module supports the CancelReceive API or not. Value true the API is supported.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
PduR	PduR/PduRBswModules
BSW Parameter	**BSW Type**
PduRCancelTransmit	EcucBooleanParamDef
BSW Description	
Specifies if the BSW module supports the CancelTransmit API or not. Value true the API is supported.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
PduR	PduR/PduRBswModules
BSW Parameter	**BSW Type**
PduRChangeParameterApi	EcucBooleanParamDef
BSW Description	
This parameter, if set to true, enables the PduR_<Up>ChangeParameter Api for this Module.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
PduR	PduR/PduRBswModules

BSW Parameter		BSW Type	
PduRCommunicationInterface		EcucBooleanParamDef	
BSW Description			
Specifies if the BSW module supports the Communication Interface APIs or not. Value true the APIs are supported. A module can have both Communication Interface APIs and Transport Protocol APIs (e.g. the COM module).			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
PduR	PduR/PduRBswModules		
BSW Parameter		**BSW Type**	
PduRLowerModule		EcucBooleanParamDef	
BSW Description			
The PduRLowerModule will decide who will call the APIs and who will implement the APIs. For example, if the CanIf module is referenced then the PDU Router module will implement the PduR_CanIfRxIndication API. And the PDUR module will call the CanIf_Transmit API. Other APIs are of course also covered. An upper module can also be an lower module (e.g. the IpduM module).			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
PduR	PduR/PduRBswModules		
BSW Parameter		**BSW Type**	
PduRRetransmission		EcucBooleanParamDef	
BSW Description			
If set to true this means that the destination transport protocol module will use the retransmission feature. This parameter might be set to false if the retransmission feature is not used, even though the destination transport protocol is supporting it. This parameter is only valid for transport protocol modules and gateway operations. If transmission from a local upper layer module this module will handle the retransmission.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context	
PduR	PduR/PduRBswModules	
BSW Parameter		**BSW Type**
PduRTransportProtocol		EcucBooleanParamDef
BSW Description		
The PDU Router module shall use the API parameters specified for transport protocol interface.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
PduR	PduR/PduRBswModules	
BSW Parameter		**BSW Type**
PduRTriggertransmit		EcucBooleanParamDef
BSW Description		
Specifies if the BSW module supports the TriggerTransmit API or not. Value true means that the BSW module supports the TriggerTransmit interface which a lower layer module can call and also that it can call the TriggerTransmit interface of an upper layer module. Value false means that the BSW module does not support the TriggerTransmit interface which a lower layer module can call and also that it shall not call the TriggerTransmit interface of an upper layer module.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
PduR	PduR/PduRBswModules	
BSW Parameter		**BSW Type**
PduRTxConfirmation		EcucBooleanParamDef
BSW Description		
Specifies if the BSW module supports the TxConfirmation API or not. Value true the API is supported.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
PduR	PduR/PduRBswModules	
BSW Parameter		**BSW Type**
PduRUpperModule		EcucBooleanParamDef
BSW Description		

The PduRUpperModule will decide who will call the APIs and who will implement the APIs. For example, if the COM module is referenced then the PDU Router module will implement the PduR_Transmit API. And the PDUR module will call the Com_RxIndication API. Other APIs are of course also covered. An upper module can also be an lower module (e.g. the IpduM module).

M2 Template	**M2 Description**

M2 Parameter

Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
PduR	PduR/PduRBswModules

BSW Parameter	**BSW Type**
PduRUseTag	EcucBooleanParamDef

BSW Description
This parameter, if set to true, enables the usage of the tag (<up>) in the following API calls: * PduR_<Up>CancelReceive * PduR_<Up>CancelTransmit * PduR_<Up>ChangeParameter Example: If used by COM and the parameter is enabled the PduR_ComCancelTransmit is used. The background is that upper layer modules differ in usage of this tag (e.g. COM is using the tag, DCM is not).

M2 Template	**M2 Description**

M2 Parameter

Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
PduR	PduR

BSW Parameter	**BSW Type**
PduRGeneral	EcucParamConfContainerDef

BSW Description
This container is a subcontainer of PduR and specifies the general configuration parameters of the PDU Router.

M2 Template	**M2 Description**

M2 Parameter

Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**

PduR	PduR/PduRGeneral		
BSW Parameter		**BSW Type**	
PduRDevErrorDetect		EcucBooleanParamDef	
BSW Description			
If true then PDU Router will enable the error-reporting to the Development Error Tracer (DET).			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
PduR	PduR/PduRGeneral		
BSW Parameter		**BSW Type**	
PduRMetaDataSupport		EcucBooleanParamDef	
BSW Description			
Enable support for MetaData handling. The size of the MetaData is defined by the configuration parameter MetaDataLength of the global PDU definitions. This feature may be used for efficient CAN-CAN-routing, where the MetaData contains the CAN ID.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
PduR	PduR/PduRGeneral		
BSW Parameter		**BSW Type**	
PduRVersionInfoApi		EcucBooleanParamDef	
BSW Description			
If true the PduR_GetVersionInfo API is available.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**	
PduR	PduR/PduRGeneral	
BSW Parameter		**BSW Type**
PduRZeroCostOperation		EcucBooleanParamDef
BSW Description		
If set the PduR configuration generator will report an error if zero-cost-operation cannot be fulfilled. This parameter shall be seen as an input requirement to the configuration generator.		
M2 Template	**M2 Description**	
M2 Parameter		

Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**	
PduR	PduR	
BSW Parameter		**BSW Type**
PduRRoutingTables		EcucParamConfContainerDef
BSW Description		
Represents one table of routing paths. This routing table allows multiple configurations that can be used to create several routing tables in the same configuration. This is mainly used for post-build (e.g. post-build selectable) but can be used by pre-compile and link-time for variant handling.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
PduR	PduR/PduRRoutingTables	
BSW Parameter		**BSW Type**
PduRConfigurationId		EcucIntegerParamDef
BSW Description		
Identification of the configuration of the PduR configuration. This identification can be read using the PduR API.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
PduR	PduR/PduRRoutingTables	
BSW Parameter		**BSW Type**
PduRMaxRoutingPathCnt		EcucIntegerParamDef
BSW Description		
Maximum number of RoutingPaths in all RoutingTables. This parameter is needed only in case of post-build loadable implementation using static memory allocation.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
PduR	PduR/PduRRoutingTables	
BSW Parameter		**BSW Type**
PduRMaxRoutingPathGroupCnt		EcucIntegerParamDef
BSW Description		
Maximum number of RoutingPathGroups. This parameter is needed only in case of post-build loadable implementation using static memory allocation.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
PduR	PduR/PduRRoutingTables	
BSW Parameter		**BSW Type**
PduRMaxRoutingTableCnt		EcucIntegerParamDef
BSW Description		
Maximum number of RoutingTables. This parameter is needed only in case of post-build loadable implementation using static memory allocation.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
PduR	PduR/PduRRoutingTables	
BSW Parameter		**BSW Type**
PduRRoutingPathGroup		EcucParamConfContainerDef
BSW Description		
This container groups routing path destinations. Destinations are used instead of routing paths since a routing path can be 1:n. It is desirable to be able to enable/disable a specific bus (i.e. a destination) rather than a routing path. Of course it is possible to create groups that covers specific routing paths as well. Enabling and disabling of routing path groups are made using the PduR API		
M2 Template	**M2 Description**	
TPS_SYST	The AUTOSAR PduR will enable and disable the sending of configurable groups of I-Pdus during runtime according to the AUTOSAR PduR specification.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::PdurIPduGroup		
Mapping Rule		**Mapping Type**
Create container for each existing PduRIPduGroup that is connected to the regarded Ecu		full

BSW Module	BSW Context	
PduR	PduR/PduRRoutingTables/PduRRoutingPathGroup	
BSW Parameter		**BSW Type**

PduRDestPduRef		EcucReferenceDef	
BSW Description			
This reference selects one destination of the routing path.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
PduR	PduR/PduRRoutingTables/PduRRoutingPathGroup		
BSW Parameter		**BSW Type**	
PduRIsEnabledAtInit		EcucBooleanParamDef	
BSW Description			
If set to true this routing path group will be enabled after initializing the PDU Router module (i.e. enabled in the PduR_Init function).			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
PduR	PduR/PduRRoutingTables/PduRRoutingPathGroup		
BSW Parameter		**BSW Type**	
PduRRoutingPathGroupId		EcucIntegerParamDef	
BSW Description			
Identification of the routing group. The identification will be used by the disable/enable API in the PDU Router module API.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**	
PduR	PduR/PduRRoutingTables	
BSW Parameter		**BSW Type**
PduRRoutingTable		EcucParamConfContainerDef
BSW Description		
Represents one container of routing paths.		
M2 Template	**M2 Description**	
TPS_SYST		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Multiplatform::IPduMapping,SystemTemplate::Fibex::FibexCore::CoreCommunication::PduTriggering,SystemTemplate::TransportProtocols::TpConfig		

<table>
<tr><th>Mapping Rule</th><th>Mapping Type</th></tr>
<tr><td>For each MultiplatformGateway.pduMapping; for each SignalPdu-Multiplexed Pdu Connection; for each IPduTriggering; for each TpConfig create one PduRRoutingPath.</td><td>full</td></tr>
</table>

<table>
<tr><th>BSW Module</th><th colspan="2">BSW Context</th></tr>
<tr><td>PduR</td><td colspan="2">PduR/PduRRoutingTables/PduRRoutingTable</td></tr>
<tr><th colspan="2">BSW Parameter</th><th>BSW Type</th></tr>
<tr><td colspan="2">PduRRoutingPath</td><td>EcucParamConfContainerDef</td></tr>
<tr><th colspan="3">BSW Description</th></tr>
<tr><td colspan="3">This container is a subcontainer of PduRRoutingTable and specifies the routing path of a PDU.</td></tr>
<tr><th>M2 Template</th><th colspan="2">M2 Description</th></tr>
<tr><td>TPS_SYST</td><td colspan="2"></td></tr>
<tr><th colspan="3">M2 Parameter</th></tr>
<tr><td colspan="3">SystemTemplate::Fibex::Fibex4Multiplatform::IPduMapping,SystemTemplate::Fibex::FibexCore::CoreCommunication::PduTriggering,SystemTemplate::TransportProtocols::TpConfig</td></tr>
<tr><th colspan="2">Mapping Rule</th><th>Mapping Type</th></tr>
<tr><td colspan="2">For each MultiplatformGateway.pduMapping; for each SignalPdu-Multiplexed Pdu Connection; for each IPduTriggering; for each TpConfig create one PduRRoutingPath.</td><td>full</td></tr>
</table>

<table>
<tr><th>BSW Module</th><th colspan="2">BSW Context</th></tr>
<tr><td>PduR</td><td colspan="2">PduR/PduRRoutingTables/PduRRoutingTable/PduRRoutingPath</td></tr>
<tr><th colspan="2">BSW Parameter</th><th>BSW Type</th></tr>
<tr><td colspan="2">PduRDestPdu</td><td>EcucParamConfContainerDef</td></tr>
<tr><th colspan="3">BSW Description</th></tr>
<tr><td colspan="3">This container is a subcontainer of PduRRoutingPath and specifies one destination for the PDU to be routed.</td></tr>
<tr><th>M2 Template</th><th colspan="2">M2 Description</th></tr>
<tr><td></td><td colspan="2"></td></tr>
<tr><th colspan="3">M2 Parameter</th></tr>
<tr><td colspan="3"></td></tr>
<tr><th colspan="2">Mapping Rule</th><th>Mapping Type</th></tr>
<tr><td colspan="2"></td><td>local</td></tr>
</table>

<table>
<tr><th>BSW Module</th><th colspan="2">BSW Context</th></tr>
<tr><td>PduR</td><td colspan="2">PduR/PduRRoutingTables/PduRRoutingTable/PduRRoutingPath/PduRDestPdu</td></tr>
<tr><th colspan="2">BSW Parameter</th><th>BSW Type</th></tr>
<tr><td colspan="2">PduRDefaultValue</td><td>EcucParamConfContainerDef</td></tr>
<tr><th colspan="3">BSW Description</th></tr>
<tr><td colspan="3">Specifies the default value of the I-PDU. Only required for gateway operation and if at least one PDU specified by PduRDestPdu uses TriggerTransmit Data provision.

Represented as an array of IntegerParamDef.</td></tr>
<tr><th>M2 Template</th><th colspan="2">M2 Description</th></tr>
<tr><td>TPS_SYST</td><td colspan="2">Default Value which will be distributed if no pdu has been received since last</td></tr>
<tr><th colspan="3">M2 Parameter</th></tr>
<tr><td colspan="3">SystemTemplate::Fibex::Fibex4Multiplatform::IPduMapping::PduMappingDefaultValue</td></tr>
<tr><th colspan="2">Mapping Rule</th><th>Mapping Type</th></tr>
</table>

Container should be created if PduMappingDefaulValue is described in the Sys-T	full

BSW Module	BSW Context	
PduR	PduR/PduRRoutingTables/PduRRoutingTable/PduRRoutingPath/PduRDestPdu/PduRDefaultValue	
BSW Parameter		**BSW Type**
PduRDefaultValueElement		EcucParamConfContainerDef
BSW Description		
Each value element is represented by the element and the position in an array.		
M2 Template	**M2 Description**	
TPS_SYST	The default value consists of a number of elements. Each element is one byte long and the number of elements is specified by SduLength.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Multiplatform::DefaultValueElement		
Mapping Rule		**Mapping Type**
Container must be created for each DefaultValueElement that is aggregated by PduMappingDefaultValue		full

BSW Module	BSW Context	
PduR	PduR/PduRRoutingTables/PduRRoutingTable/PduRRoutingPath/PduRDestPdu/PduRDefaultValue/PduRDefaultValueElement	
BSW Parameter		**BSW Type**
PduRDefaultValueElement		EcucIntegerParamDef
BSW Description		
The default value consists of a number of elements. Each element is one byte long and the number of elements is specified by SduLength. The position of this parameter in the container is specified by the PduRElementBytePosition parameter.		
M2 Template	**M2 Description**	
TPS_SYST	The integer value of a freely defined data byte.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Multiplatform::DefaultValueElement.elementByteValue		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
PduR	PduR/PduRRoutingTables/PduRRoutingTable/PduRRoutingPath/PduRDestPdu/PduRDefaultValue/PduRDefaultValueElement	
BSW Parameter		**BSW Type**
PduRDefaultValueElementBytePosition		EcucIntegerParamDef
BSW Description		
This parameter specifies the byte position of the element within the default value		
M2 Template	**M2 Description**	
TPS_SYST	The default value consists of a number of elements. Each element is one byte long and the number of elements is specified by SduLength.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Multiplatform::DefaultValueElement..elementPosition		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context
PduR	PduR/PduRRoutingTables/PduRRoutingTable/PduRRoutingPath/PduRDestPdu
BSW Parameter	**BSW Type**
PduRDestPduDataProvision	EcucEnumerationParamDef
BSW Description	
Specifies how data are provided: direct (as part of the Transmit call) or via the TriggerTransmit callback function. Only required for non-TP gatewayed I-PDUs.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context
PduR	PduR/PduRRoutingTables/PduRRoutingTable/PduRRoutingPath/PduRDestPdu
BSW Parameter	**BSW Type**
PduRDestPduHandleId	EcucIntegerParamDef
BSW Description	
PDU identifier assigned by PDU Router. Used by communication interface and transport protocol modules for confirmation (PduR_<Lo>TxConfirmation) and for TriggerTransmit (PduR_<Lo>TriggerTransmit).	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context
PduR	PduR/PduRRoutingTables/PduRRoutingTable/PduRRoutingPath/PduRDestPdu
BSW Parameter	**BSW Type**
PduRDestPduRef	EcucReferenceDef
BSW Description	
Destination PDU reference; reference to unique PDU identifier which shall be used by the PDU Router instead of the source PDU ID when calling the related function of the destination module.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context
PduR	PduR/PduRRoutingTables/PduRRoutingTable/PduRRoutingPath/PduRDestPdu
BSW Parameter	**BSW Type**

PduRDestTxBufferRef		EcucReferenceDef	
BSW Description			
Reference to a buffer that is allocated in the PduRTxBufferTable. This buffer is required for communication interface gatewaying, and for transport protocol gatewaying for single frame routing.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
PduR	PduR/PduRRoutingTables/PduRRoutingTable/PduRRoutingPath/PduRDestPdu		
BSW Parameter		**BSW Type**	
PduRTpThreshold		EcucIntegerParamDef	
BSW Description			
Defines the number of bytes which shall be received before transmission on the destination bus may start. Only required for routing-on-the-fly TP gateway PDUs. The threshold shall not be larger than the length of the related TP Buffer.			
M2 Template	**M2 Description**		
TPS_SYST	Optionally defines the to be configured Pdu Router TpChunkSize for		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Multiplatform::IPduMapping.pdurTpChunkSize			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	**BSW Context**		
PduR	PduR/PduRRoutingTables/PduRRoutingTable/PduRRoutingPath/PduRDestPdu		
BSW Parameter		**BSW Type**	
PduRTransmissionConfirmation		EcucBooleanParamDef	
BSW Description			
This parameter is only for communication interfaces. Transport protocol modules will always call the TxConfirmation function. If set the destination communication interface module will call the TxConfirmation. However the TxConfirmation may be not called due to error. So the PduR shall not block until the TxConfirmation is called. One background for this parameter is for the PduR to know when all modules have confirmed a multicast operation.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**
PduR	PduR/PduRRoutingTables/PduRRoutingTable/PduRRoutingPath

BSW Parameter	BSW Type
PduRSrcPdu	EcucParamConfContainerDef
BSW Description	
This container is a subcontainer of PduRRoutingPath and specifies the source of the PDU to be routed.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context
PduR	PduR/PduRRoutingTables/PduRRoutingTable/PduRRoutingPath/PduRSrcPdu
BSW Parameter	**BSW Type**
PduRSourcePduHandleId	EcucIntegerParamDef
BSW Description	
PDU identifier assigned by PDU Router.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context
PduR	PduR/PduRRoutingTables/PduRRoutingTable/PduRRoutingPath/PduRSrcPdu
BSW Parameter	**BSW Type**
PduRSrcPduRef	EcucReferenceDef
BSW Description	
Source PDU reference; reference to unique PDU identifier which shall be used for the requested PDU Router operation.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context
PduR	PduR/PduRRoutingTables/PduRRoutingTable/PduRRoutingPath/PduRSrcPdu
BSW Parameter	**BSW Type**
PduRSrcPduUpTxConf	EcucBooleanParamDef
BSW Description	
When enabled, the TxConfirmation will be forwarded to the upper layer. Prerequisites: Lower layer and upper layer support TxConfirmation.	
M2 Template	**M2 Description**
M2 Parameter	

Mapping Rule	Mapping Type

BSW Module	BSW Context		
PduR	PduR/PduRRoutingTables		
BSW Parameter		**BSW Type**	
PduRTpBufferTable		EcucParamConfContainerDef	
BSW Description			
This container will specify the needed buffers for gatewaying using TP. It is not connected to the specific routing path destination to allow a more efficient buffer handling.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
PduR	PduR/PduRRoutingTables/PduRTpBufferTable		
BSW Parameter		**BSW Type**	
PduRMaxTpBufferNumber		EcucIntegerParamDef	
BSW Description			
maximum number of TP buffers. This parameter is obsolete and will be removed in future.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
PduR	PduR/PduRRoutingTables/PduRTpBufferTable		
BSW Parameter		**BSW Type**	
PduRTpBuffer		EcucParamConfContainerDef	
BSW Description			
Specifies a buffer used for gatwaying through TP.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context	
PduR	PduR/PduRRoutingTables/PduRTpBufferTable/PduRTpBuffer	
BSW Parameter		**BSW Type**
PduRTpBufferLength		EcucIntegerParamDef

BSW Description		
Length of the TP buffer in number of bytes		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context		
PduR	PduR/PduRRoutingTables		
BSW Parameter		**BSW Type**	
PduRTxBufferTable		EcucParamConfContainerDef	
BSW Description			
This container contains the buffers used for gatewaying via communication interfaces and for single frames of transport protocols.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
PduR	PduR/PduRRoutingTables/PduRTxBufferTable		
BSW Parameter		**BSW Type**	
PduRMaxTxBufferNumber		EcucIntegerParamDef	
BSW Description			
maximum number of Tx buffers. This parameter is set to obsolete and will be removed in future.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
PduR	PduR/PduRRoutingTables/PduRTxBufferTable		
BSW Parameter		**BSW Type**	
PduRTxBuffer		EcucParamConfContainerDef	
BSW Description			
Specifies a buffer used for gatewaying via communication interfaces or for single frames of transport protocols.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

	local

BSW Module	BSW Context	
PduR	PduR/PduRRoutingTables/PduRTxBufferTable/PduRTxBuffer	
BSW Parameter		**BSW Type**
PduRPduMaxLength		EcucIntegerParamDef
BSW Description		
Length of the Tx buffer in bytes. This parameter limits the size of buffered routed PDUs.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
PduR	PduR/PduRRoutingTables/PduRTxBufferTable/PduRTxBuffer	
BSW Parameter		**BSW Type**
PduRTxBufferDepth		EcucIntegerParamDef
BSW Description		
Number of Pdus that can be stored in the buffer. If the value is 1, the buffer semantic is "last is best". If the value is greater than 1, the buffer semantic is a FiFo. For TP single frames, the depth is always 1.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

D.1.4 Nm Interface

BSW Module	BSW Context	
Nm	Nm	
BSW Parameter		**BSW Type**
NmChannelConfig		EcucParamConfContainerDef
BSW Description		
This container contains the configuration (parameters) of the bus channel(s). The channel parameter shall be harmonized within the whole communication stack.		
M2 Template	**M2 Description**	
TPS_SYST	Set of NM nodes coordinated with use of the NM algorithm.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmCluster		
Mapping Rule		**Mapping Type**
Create Container for each existing NmCluster.		full

BSW Module	BSW Context
Nm	Nm/NmChannelConfig

BSW Parameter		BSW Type
NmActiveCoordinator		EcucBooleanParamDef
BSW Description		
This parameter indicates whether a NM channel - part of a Nm Coordination cluster - will be coordinated actively (NmActiveCoordinator = TRUE) or passively (NmActiveCoordinator = FALSE).		
M2 Template	**M2 Description**	
TPS_SYST	This attribute indicates whether a NM Coordinator is an active gateway (true) or a passive gateway (false).	
M2 Parameter		
SystemTemplate::NetworkManagement::NmCoordinator.nmActiveCoordinator		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Nm	Nm/NmChannelConfig	
BSW Parameter		**BSW Type**
NmBusType		EcucChoiceContainerDef
BSW Description		
M2 Template	**M2 Description**	
TPS_SYST		
M2 Parameter		
SystemTemplate::NetworkManagement::CanNmCluster,SystemTemplate::NetworkManagement::FlexrayNmCluster,SystemTemplate::NetworkManagement::UdpNmCluster		
Mapping Rule		**Mapping Type**
Bus Type can be derived from the BusNm Configuration in the System Description.		full

BSW Module	BSW Context	
Nm	Nm/NmChannelConfig/NmBusType	
BSW Parameter		**BSW Type**
NmGenericBusNmConfig		EcucParamConfContainerDef
BSW Description		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
Nm	Nm/NmChannelConfig/NmBusType/NmGenericBusNmConfig	
BSW Parameter		**BSW Type**
NmGenericBusNmPrefix		EcucStringParamDef
BSW Description		
The prefix which identifies the generic <BusNm>. This will be used to determine the API name to be called by Nm for the provided interfaces of the <BusNm>. This string will used for the module prefix before the "_" character in the API call name.		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**
Nm	Nm/NmChannelConfig/NmBusType/NmGenericBusNmConfig
BSW Parameter	**BSW Type**
NmGenericBusNmShutdownTime	EcucFloatParamDef
BSW Description	
This parameter shall be used to calculate shutdown delay time.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**
Nm	Nm/NmChannelConfig/NmBusType
BSW Parameter	**BSW Type**
NmStandardBusNmConfig	EcucParamConfContainerDef
BSW Description	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**
Nm	Nm/NmChannelConfig/NmBusType/NmStandardBusNmConfig
BSW Parameter	**BSW Type**
NmStandardBusType	EcucEnumerationParamDef
BSW Description	
Identifies the bus type of the channel for standard AUTOSAR <BusNm>s and is used to determine which set of API calls to be called by Nm for the <BusNm>s. Note: The Ethernet bus' NM is UdpNm !	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**

Nm	Nm/NmChannelConfig
BSW Parameter	**BSW Type**
NmChannelId	EcucIntegerParamDef
BSW Description	
This parameter is obsolete since it is not needed anymore, information is given by NmComMChannelRef. Old description: This parameter holds the unique channel index value. The value shall be the same as the ComMChannelId of the ComMChannel referenced by NmComMChannelRef. Implementation Type: NetworkHandleType	
M2 Template	**M2 Description**
TPS_SYST	Channel identification number of the corresponding channel. Must be unique over all NmClusters.
M2 Parameter	
SystemTemplate::NetworkManagement::NmCluster.nmChannelId	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
Nm	Nm/NmChannelConfig
BSW Parameter	**BSW Type**
NmChannelSleepMaster	EcucBooleanParamDef
BSW Description	
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. If this parameter is set to TRUE, the Nm shall assume that the channel is always ready to go to sleep and that no callouts to Nm_RemoteSleepIndication or Nm_RemoteSleepCancellation will be made from the <BusNm> representing this channel. If this parameter is set to FALSE, the Nm shall not assume that the network is ready to sleep until a callout has been made to Nm_RemoteSleepCancellation.	
M2 Template	**M2 Description**
TPS_SYST	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.
M2 Parameter	
SystemTemplate::NetworkManagement::NmCluster.nmChannelSleepMaster	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
Nm	Nm/NmChannelConfig
BSW Parameter	**BSW Type**
NmComMChannelRef	EcucSymbolicNameReferenceDef
BSW Description	
Reference to the corresponding ComM Channel.	
M2 Template	**M2 Description**
M2 Parameter	

Mapping Rule	Mapping Type
	local

BSW Module	BSW Context	
Nm	Nm/NmChannelConfig	
BSW Parameter		**BSW Type**
NmCoordClusterIndex		EcucIntegerParamDef
BSW Description		
If this parameter is undefined for a channel, the corresponding bus does not belong to an NM coordination cluster.		
M2 Template	**M2 Description**	
TPS_SYST	Identification of the NMCoordinator.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmCoordinator.index		
Mapping Rule		**Mapping Type**
Search for NmCoordinator element that contains a reference to the regarded channel.		full

BSW Module	BSW Context	
Nm	Nm/NmChannelConfig	
BSW Parameter		**BSW Type**
NmShutdownDelayTimer		EcucFloatParamDef
BSW Description		
This parameter defines the time in seconds which the NM Coordination algorithm shall delay the release of this channel with. This parameter is obsolete since it is not needed anymore as delay can be always calculated.		
M2 Template	**M2 Description**	
TPS_SYST	This parameter defines the time in seconds which the NM Coordination algorithm shall delay the release of the referenced cluster.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmCoordinator.nmShutdownDelayTimer		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Nm	Nm/NmChannelConfig	
BSW Parameter		**BSW Type**
NmStateReportEnabled		EcucBooleanParamDef
BSW Description		
Specifies if the NMS shall be set for the corresponding network. false: No NMS shall be set true: The NMS shall be set		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Nm	Nm/NmChannelConfig	
BSW Parameter		**BSW Type**
NmStateReportSignalRef		EcucSymbolicNameReferenceDef
BSW Description		
Reference to the signal for setting the NMS by calling Com_SendSignal for the respective channel.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Nm	Nm/NmChannelConfig	
BSW Parameter		**BSW Type**
NmSynchronizingNetwork		EcucBooleanParamDef
BSW Description		
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.		
M2 Template	**M2 Description**	
TPS_SYST	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.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmCluster.nmSynchronizingNetwork		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Nm	Nm	
BSW Parameter		**BSW Type**
NmGlobalConfig		EcucParamConfContainerDef
BSW Description		
This container contains all global configuration parameters of the Nm Interface.		
M2 Template	**M2 Description**	
M2 Parameter		
SystemTemplate::NetworkManagement::NmCoordinator		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Nm	Nm/NmGlobalConfig	
BSW Parameter		**BSW Type**
NmGlobalConstants		EcucParamConfContainerDef
BSW Description		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context	
Nm	Nm/NmGlobalConfig/NmGlobalConstants	
BSW Parameter		**BSW Type**
NmNumberOfChannels		EcucIntegerParamDef
BSW Description		
Number of NM channels allowed within one ECU.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Nm	Nm/NmGlobalConfig	
BSW Parameter		**BSW Type**
NmGlobalFeatures		EcucParamConfContainerDef
BSW Description		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Nm	Nm/NmGlobalConfig/NmGlobalFeatures	
BSW Parameter		**BSW Type**
NmBusSynchronizationEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling bus synchronization support of the <BusNm>s. This feature is required for NM Coordinator nodes only.		
M2 Template	**M2 Description**	
TPS_SYST	Enables bus synchronization support.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmBusSynchronizationEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Nm	Nm/NmGlobalConfig/NmGlobalFeatures	
BSW Parameter		**BSW Type**
NmCarWakeUpCallback		EcucFunctionNameDef

BSW Description		
Name of the callback function to be called if Nm_CarWakeUpIndication() is called.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context		
Nm	Nm/NmGlobalConfig/NmGlobalFeatures		
BSW Parameter		**BSW Type**	
NmCarWakeUpRxEnabled		EcucBooleanParamDef	
BSW Description			
Enables or disables CWU detection. FALSE - CarWakeUp not supported TRUE - CarWakeUp supported			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
Nm	Nm/NmGlobalConfig/NmGlobalFeatures		
BSW Parameter		**BSW Type**	
NmComControlEnabled		EcucBooleanParamDef	
BSW Description			
Pre-processor switch for enabling the Communication Control support.			
M2 Template	**M2 Description**		
TPS_SYST	Enables the Communication Control support.		
M2 Parameter			
SystemTemplate::NetworkManagement::NmEcu.nmComControlEnabled			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	BSW Context		
Nm	Nm/NmGlobalConfig/NmGlobalFeatures		
BSW Parameter		**BSW Type**	
NmComUserDataSupport		EcucBooleanParamDef	
BSW Description			
Enable/Disable setting of NMUserData via SW-C. If NmComUserDataSupport is enabled the API Nm_SetUserData shall not be available.			
M2 Template	**M2 Description**		
TPS_SYST	Switch for enabling user data support.		
M2 Parameter			
SystemTemplate::NetworkManagement::NmEcu.nmUserDataEnabled			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	BSW Context	
Nm	Nm/NmGlobalConfig/NmGlobalFeatures	
BSW Parameter		**BSW Type**
NmCoordinatorSupportEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling NM Coordinator support.		
M2 Template	**M2 Description**	
TPS_SYST	Nm ECU may coordinate different clusters.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmCoordinator		
Mapping Rule		**Mapping Type**
If NmCoordinators are defined set this parameter to true.		full

BSW Module	BSW Context	
Nm	Nm/NmGlobalConfig/NmGlobalFeatures	
BSW Parameter		**BSW Type**
NmCoordinatorSyncSupport		EcucBooleanParamDef
BSW Description		
Enables/disables the coordinator synchronisation support.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
Nm	Nm/NmGlobalConfig/NmGlobalFeatures	
BSW Parameter		**BSW Type**
NmGlobalCoordinatorTime		EcucFloatParamDef
BSW Description		
This parameter defines the maximum shutdown time of a connected and coordinated NM-Cluster. Note:This includes nested connections.		
M2 Template	**M2 Description**	
TPS_SYST	This attribute defines the maximum shutdown time (in seconds) of a connected and coordinated NM-Cluster.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmCoordinator.nmGlobalCoordinatorTime		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Nm	Nm/NmGlobalConfig/NmGlobalFeatures	
BSW Parameter		**BSW Type**
NmNodeDetectionEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling the Node Detection feature.		
M2 Template	**M2 Description**	
TPS_SYST	Enables the Request Repeat Message Request support. Only valid if nmNodeIdEnabled is set to true.	

M2 Parameter	
SystemTemplate::NetworkManagement::NmEcu.nmNodeDetectionEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	BSW Context	
Nm	Nm/NmGlobalConfig/NmGlobalFeatures	
BSW Parameter		**BSW Type**
NmNodeIdEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling transmission of the source node identifier in NM messages.		
M2 Template	**M2 Description**	
TPS_SYST	Enables the source node identifier.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmNodeIdEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Nm	Nm/NmGlobalConfig/NmGlobalFeatures	
BSW Parameter		**BSW Type**
NmPassiveModeEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling support of Passive Mode of the <BusNm>s.		
M2 Template	**M2 Description**	
TPS_SYST	Enables support of the Passive Mode.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmPassiveModeEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Nm	Nm/NmGlobalConfig/NmGlobalFeatures	
BSW Parameter		**BSW Type**
NmPduRxIndicationEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling the PDU Rx Indication.		
M2 Template	**M2 Description**	
TPS_SYST	Switch for enabling the PDU Rx Indication.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmPduRxIndicationEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Nm	Nm/NmGlobalConfig/NmGlobalFeatures	
BSW Parameter		**BSW Type**
NmRemoteSleepIndEnabled		EcucBooleanParamDef
BSW Description		

Pre-processor switch for enabling Remote Sleep Indication support. This feature is required for a Gateway or Nm Coordinator functionality.		
M2 Template	**M2 Description**	
TPS_SYST	Switch for enabling remote sleep indication support.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmRemoteSleepIndEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Nm	Nm/NmGlobalConfig/NmGlobalFeatures	
BSW Parameter		**BSW Type**
NmRepeatMsgIndEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling the Repeat Message Bit Indication.		
M2 Template	**M2 Description**	
TPS_SYST	Switch for enabling the Repeat Message Bit Indication.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmRepeatMsgIndEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Nm	Nm/NmGlobalConfig/NmGlobalFeatures	
BSW Parameter		**BSW Type**
NmStateChangeIndEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling the Network Management state change notification.		
M2 Template	**M2 Description**	
TPS_SYST	Enables the CAN Network Management state change notification.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmStateChangeIndEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Nm	Nm/NmGlobalConfig/NmGlobalFeatures	
BSW Parameter		**BSW Type**
NmUserDataEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling User Data support.		
M2 Template	**M2 Description**	
TPS_SYST	Switch for enabling user data support.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmUserDataEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**

Nm	Nm/NmGlobalConfig		
BSW Parameter		**BSW Type**	
NmGlobalProperties		EcucParamConfContainerDef	
BSW Description			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
Nm	Nm/NmGlobalConfig/NmGlobalProperties		
BSW Parameter		**BSW Type**	
NmCycletimeMainFunction		EcucFloatParamDef	
BSW Description			
The period between successive calls to the Main Function of the NM Interface in seconds.			
M2 Template	**M2 Description**		
TPS_SYST	The period between successive calls to the Main Function of the NM Interface in seconds.		
M2 Parameter			
SystemTemplate::NetworkManagement::NmEcu.nmCycletimeMainFunction			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	**BSW Context**		
Nm	Nm/NmGlobalConfig/NmGlobalProperties		
BSW Parameter		**BSW Type**	
NmDevErrorDetect		EcucBooleanParamDef	
BSW Description			
Pre-processor switch for enabling development error detection and notification.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
Nm	Nm/NmGlobalConfig/NmGlobalProperties		
BSW Parameter		**BSW Type**	
NmVersionInfoApi		EcucBooleanParamDef	
BSW Description			
Pre-processor switch for enabling Version Info API support.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

	local

D.1.5 EcuC

BSW Module	BSW Context		
EcuC	EcuC		
BSW Parameter		**BSW Type**	
EcucConfigSet		EcucParamConfContainerDef	
BSW Description			
This container contains the configuration parameters and sub containers of the global PduCollection. This container is a MultipleConfigurationContainer, i.e. this container and its sub-containers exist once per configuration set.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	BSW Context		
EcuC	EcuC/EcucConfigSet		
BSW Parameter		**BSW Type**	
EcucPduCollection		EcucParamConfContainerDef	
BSW Description			
Collection of all Pdu objects flowing through the Com-Stack.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	BSW Context		
EcuC	EcuC/EcucConfigSet/EcucPduCollection		
BSW Parameter		**BSW Type**	
Pdu		EcucParamConfContainerDef	
BSW Description			
One Pdu flowing through the COM-Stack. This Pdu is used by all Com-Stack modules to agree on referencing the same Pdu.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	BSW Context	
EcuC	EcuC/EcucConfigSet/EcucPduCollection/Pdu	
BSW Parameter		**BSW Type**

MetaDataLength		EcucIntegerParamDef
BSW Description		
Number of additional bytes in PDU data that contain auxiliary information (MetaData) for the PDU, e.g. the CAN ID. These bytes are handled as part of the PDU data and increase the PDU data length accordingly.		
M2 Template	**M2 Description**	
TPS_SYST	Number of additional bytes of MetaData in the PDU data field. The MetaData contains auxiliary information for the PDU, e.g. the CAN ID.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::Pdu.metaDataLength		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
EcuC	EcuC/EcucConfigSet/EcucPduCollection/Pdu	
BSW Parameter		**BSW Type**
PduLength		EcucIntegerParamDef
BSW Description		
Length of the Pdu in bytes. 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.		
M2 Template	**M2 Description**	
TPS_SYST	Pdu length in bytes.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::Pdu.length		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
EcuC	EcuC/EcucConfigSet/EcucPduCollection/Pdu	
BSW Parameter		**BSW Type**
SysTPduToFrameMappingRef		EcucForeignReferenceDef
BSW Description		
Optional reference to the PduToFrameMapping from the SystemTemplate which this Pdu represents. This parameter is obsolete and will be removed in future.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
EcuC	EcuC/EcucConfigSet/EcucPduCollection/Pdu	
BSW Parameter		**BSW Type**
SysTPduToFrameTriggeringRef		EcucForeignReferenceDef
BSW Description		

Reference to the FrameTriggering from the SystemTemplate which this Pdu belongs to. SysTPduToFrameTriggeringRef shall be used for UserDefinedPdus, NmPdus and NPdus which are not going through the Pdu Router. This reference shall not be used if SysTPduToPduTriggeringRef exists.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
EcuC	EcuC/EcucConfigSet/EcucPduCollection/Pdu	
BSW Parameter		**BSW Type**
SysTPduToPduTriggeringRef		EcucForeignReferenceDef
BSW Description		
Reference to the PduTriggering from the SystemTemplate which this Pdu represents. SysTPduToPduTriggeringRef shall be used for all Pdus except UserDefinedPdus, NmPdus and NPdus which are not going through the Pdu Router. For these Pdus, SysTPduToFrameTriggeringRef shall be used.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
EcuC	EcuC/EcucConfigSet/EcucPduCollection	
BSW Parameter		**BSW Type**
PduIdTypeEnum		EcucEnumerationParamDef
BSW Description		
The PduIdType is used within the entire AUTOSAR Com Stack except for bus drivers. The size of this global type depends on the maximum number of PDUs used within one software module. If no software module deals with more PDUs that 256, this type can be set to uint8. If at least one software module handles more than 256 PDUs, this type must be set to uint16. See AUTOSAR_SWS_CommunicationStackTypes for more details.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
EcuC	EcuC/EcucConfigSet/EcucPduCollection	
BSW Parameter		**BSW Type**
PduLengthTypeEnum		EcucEnumerationParamDef
BSW Description		

The PduLengthType is used within the entire AUTOSAR Com Stack except for bus drivers. The size of this global type depends on the maximum length of PDUs to be sent by an ECU. If no segmentation is used the length depends on the maximum payload size of a frame of the underlying communication system (for FlexRay maximum size is 255 bytes, therefore uint8). If segmentation is used it depends on the maximum length of a segmented N-PDU (in general uint16 is used). See AUTOSAR_SWS_CommunicationStackTypes for more details.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
EcuC	EcuC	
BSW Parameter		**BSW Type**
EcucHardware		EcucParamConfContainerDef
BSW Description		
Hardware definition of this Ecu.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
EcuC	EcuC/EcucHardware	
BSW Parameter		**BSW Type**
EcucCoreDefinition		EcucParamConfContainerDef
BSW Description		
Definition of one Core on this Ecu.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
EcuC	EcuC/EcucHardware/EcucCoreDefinition	
BSW Parameter		**BSW Type**
EcucCoreHwRef		EcucForeignReferenceDef
BSW Description		
Optional reference to the HwElement of HwCategory ProcessingUnit that represents this Core in the ECU Resource Template.		
M2 Template	**M2 Description**	
M2 Parameter		

Mapping Rule	Mapping Type

BSW Module	BSW Context	
EcuC	EcuC/EcucHardware/EcucCoreDefinition	
BSW Parameter		**BSW Type**
EcucCoreId		EcucIntegerParamDef
BSW Description		
ID of the core.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		ECU

BSW Module	BSW Context	
EcuC	EcuC	
BSW Parameter		**BSW Type**
EcucPartitionCollection		EcucParamConfContainerDef
BSW Description		
Collection of Partitions defined for this ECU.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
EcuC	EcuC/EcucPartitionCollection	
BSW Parameter		**BSW Type**
EcucPartition		EcucParamConfContainerDef
BSW Description		
Definition of one Partition on this ECU. One Partition will be implemented using one Os-Application.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
EcuC	EcuC/EcucPartitionCollection/EcucPartition	
BSW Parameter		**BSW Type**
EcucPartitionBswModuleDistinguishedPartition		EcucForeignReferenceDef
BSW Description		
This maps the abstract partition of the Bsw Module to a concrete Partition existing in the ECU.		

M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**
EcuC	EcuC/EcucPartitionCollection/EcucPartition
BSW Parameter	**BSW Type**
EcucPartitionBswModuleExecution	EcucBooleanParamDef
BSW Description	
Denotes that this partition will execute BSW Modules. BSW Modules can only be executed in such partitions.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**
EcuC	EcuC/EcucPartitionCollection/EcucPartition
BSW Parameter	**BSW Type**
EcucPartitionSoftwareComponentInstanceRef	EcucInstanceReferenceDef
BSW Description	
References the SW Component instances from the Ecu Extract that shall be executed in this partition.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**
EcuC	EcuC/EcucPartitionCollection/EcucPartition
BSW Parameter	**BSW Type**
PartitionCanBeRestarted	EcucBooleanParamDef
BSW Description	
Specifies the requirement whether the Partition can be restarted. If set to true all software executing in this partition shall be capable of handling a restart.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**

EcuC	EcuC		
BSW Parameter		**BSW Type**	
EcucUnitGroupAssignment		EcucParamConfContainerDef	
BSW Description			
Collection of UnitGroup references to support the generation of ASAM MCD file.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	**BSW Context**		
EcuC	EcuC/EcucUnitGroupAssignment		
BSW Parameter		**BSW Type**	
EcucUnitGroupRef		EcucForeignReferenceDef	
BSW Description			
Optional reference to the UnitGroup to support the generation of ASAM MCD file. These UnitGroups are selecting a set of units for a specific country.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	**BSW Context**		
EcuC	EcuC		
BSW Parameter		**BSW Type**	
EcucVariationResolver		EcucParamConfContainerDef	
BSW Description			
Collection of PredefinedVariant elements containing definition of values for SwSystemconst which shall be applied when resolving the variability during ECU Configuration.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	**BSW Context**	
EcuC	EcuC/EcucVariationResolver	
BSW Parameter		**BSW Type**
PredefinedVariantRef		EcucForeignReferenceDef
BSW Description		
M2 Template	**M2 Description**	
M2 Parameter		

Mapping Rule	Mapping Type

D.1.6 ComM

BSW Module	BSW Context	
ComM	ComM	
BSW Parameter		**BSW Type**
ComMConfigSet		EcucParamConfContainerDef
BSW Description		
This container is the base for a multiple configuration set.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet	
BSW Parameter		**BSW Type**
ComMChannel		EcucParamConfContainerDef
BSW Description		
This container contains the configuration (parameters) of the bus channel(s). The channel parameters shall be harmonized within the whole communication stack.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMChannel	
BSW Parameter		**BSW Type**
ComMBusType		EcucEnumerationParamDef
BSW Description		
Identifies the bus type of the channel.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMChannel	
BSW Parameter		**BSW Type**
ComMCDDBusPrefix		EcucStringParamDef

BSW Description		
Prefix to be used for API calls to CDD.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMChannel	
BSW Parameter		**BSW Type**
ComMChannelId		EcucIntegerParamDef
BSW Description		
Channel identification number of the corresponding channel.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMChannel	
BSW Parameter		**BSW Type**
ComMFullCommRequestNotificationEnabled		EcucBooleanParamDef
BSW Description		
Defines if the optional SenderReceiver Port of Interface ComM_CurrentChannelRequest will be provided for this channel. True means enabled. False means disabled		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMChannel	
BSW Parameter		**BSW Type**
ComMGlobalNvmBlockDescriptor		EcucBooleanParamDef
BSW Description		
If this parameter is set to "true", the NoWakeUp inhibition state of the channel shall be stored (in some implementation specific way) in the block pointed to by ComMGlobalNvmBlockDescriptor.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMChannel	
BSW Parameter		**BSW Type**
ComMMainFunctionPeriod		EcucFloatParamDef
BSW Description		
Specifies the period in seconds that the MainFunction has to be triggered with. Comment: ComM scheduling shall be at least as fast as the communication stack and a schedule longer than 100ms makes no sense for communication.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMChannel	
BSW Parameter		**BSW Type**
ComMNetworkManagement		EcucParamConfContainerDef
BSW Description		
This container contains the configuration parameters of the networkmanagement.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMChannel/ComMNetworkManagement	
BSW Parameter		**BSW Type**
ComMNmLightTimeout		EcucFloatParamDef
BSW Description		
Defines the timeout (in seconds) after COMM_FULL_COMMUNICATION sub-state COMM_FULL_COM_READY_SLEEP is left. The range shall be greater than 0.0 and less or equal to 255.0.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMChannel/ComMNetworkManagement	
BSW Parameter		**BSW Type**
ComMNmVariant		EcucEnumerationParamDef
BSW Description		

Defines the functionality of the networkmanagement. Shall be harmonized with NM configuration.		
M2 Template	**M2 Description**	
TPS_SYST	Asscociation of the NmNode to the CommunicationController in the topology description.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmNode.controller		
Mapping Rule		**Mapping Type**
If the CommunicationController is not referenced by NmNode the ComMNm Variant of the corresponding ComMChannel shall be set to NONE if not explicitly set to LIGHT. If the CommunicationController is referenced by NmNode where the nmPassiveModeEnabled attribute is present and is set to true, the ComMNm Variant shall be set to PASSIVE. If the CommunicationController is referenced by NmNode where the nmPassiveModeEnabled attribute is not present or is set to false the ComMNmVariant shall be set to FULL.		full

BSW Module	**BSW Context**	
ComM	ComM/ComMConfigSet/ComMChannel/ComMNetworkManagement	
BSW Parameter		**BSW Type**
ComMPncNmRequest		EcucBooleanParamDef
BSW Description		
If this parameter equals true then every time a FULL Communication is requested due to a change in the PNC state machine to PNC_REQUESTED Nm shall be called using the API Nm_NetworkRequest.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
ComM	ComM/ComMConfigSet/ComMChannel	
BSW Parameter		**BSW Type**
ComMNoCom		EcucBooleanParamDef
BSW Description		
Not allowed to change state of ComM channel to COMM_SILENT_COMMUNICATION or COMM_FULL_COMMUNICATION. true: Enabled - Not allowed to switch to Communication Modes above. false: Disabled - Allowed to switch Communication Modes above. Shall be possible to change parameter during runtime with ComM API's. ECU/All channels: ComM_LimitECUToNoComMode(). Separate channels: ComM_LimitChannelToNoComMode().		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMChannel	
BSW Parameter		**BSW Type**
ComMNoWakeup		EcucBooleanParamDef
BSW Description		
Defines if an ECU is not allowed to wake-up the channel. true: Enabled (not allowed to wake-up)) false: Disabled This is the default/init value of a runtime variable that can be changed during runtime using ComM_PreventWakeUp().		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMChannel	
BSW Parameter		**BSW Type**
ComMPncGatewayType		EcucEnumerationParamDef
BSW Description		
Identifies the Partial Network Gateway behaviour of a ComMChannel.		
M2 Template	**M2 Description**	
TPS_SYST		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreTopology::CommunicationConnector.pncGatewayType		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMChannel	
BSW Parameter		**BSW Type**
ComMUserPerChannel		EcucParamConfContainerDef
BSW Description		
This container contains a list of identifiers that are needed to refer to a user in the system which is linked to a channel.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMChannel/ComMUserPerChannel	
BSW Parameter		**BSW Type**
ComMUserChannel		EcucReferenceDef
BSW Description		

Reference to the ComMUser that corresponds to this channel user. ImplementationType: COMM_UserHandleType		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
ComM	ComM/ComMConfigSet	
BSW Parameter		**BSW Type**
ComMPnc		EcucParamConfContainerDef
BSW Description		
This container contains the configuration of the partial network cluster (PNC).		
M2 Template	**M2 Description**	
TPS_SYST	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.	
M2 Parameter		
SystemTemplate::PncMapping::PncMapping		
Mapping Rule		**Mapping Type**
Create ComMPnc container for each PncMapping element.		full

BSW Module	**BSW Context**	
ComM	ComM/ComMConfigSet/ComMPnc	
BSW Parameter		**BSW Type**
ComMChannelPerPnc		EcucReferenceDef
BSW Description		
Reference to the ComMChannel that is required for this PNC. ImplementationType: NetworkHandleType		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
ComM	ComM/ComMConfigSet/ComMPnc	
BSW Parameter		**BSW Type**
ComMPncComSignal		EcucParamConfContainerDef
BSW Description		
Represents the PncComSignals which are used to communicate the EIRA and ERA status of this PNC.		
M2 Template	**M2 Description**	
M2 Parameter		

Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**	
ComM	ComM/ComMConfigSet/ComMPnc/ComMPncComSignal	
BSW Parameter		**BSW Type**
ComMPncComSignalChannelRef		EcucReferenceDef
BSW Description		
Reference to the ComMChannel which is used to determine whether this PncComSignal shall participate in the active or passive role (via the parameter ComMPncGatewayType of the ComMChannel). This information may be available by following the ComMPncComSignalRef and analyse the Com configuration as well.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
ComM	ComM/ComMConfigSet/ComMPnc/ComMPncComSignal	
BSW Parameter		**BSW Type**
ComMPncComSignalDirection		EcucEnumerationParamDef
BSW Description		
Indicates the communication direction of this PncComSignal.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
ComM	ComM/ComMConfigSet/ComMPnc/ComMPncComSignal	
BSW Parameter		**BSW Type**
ComMPncComSignalKind		EcucEnumerationParamDef
BSW Description		
Indicates whether this PncComSignal represents EIRA or ERA PNC information. This parameter ComMPncComSignalKind is optional and shall be ignored when ComMPncComSignalDirection equals TX.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMPnc/ComMPncComSignal	
BSW Parameter		**BSW Type**
ComMPncComSignalRef		EcucSymbolicNameReferenceDef
BSW Description		
Reference to the ComSignal which is used to transport the partial network channel request information.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMPnc	
BSW Parameter		**BSW Type**
ComMPncId		EcucIntegerParamDef
BSW Description		
Partial network cluster identification number.		
M2 Template	**M2 Description**	
TPS_SYST	Identifer of the Partial Network Cluster.	
M2 Parameter		
SystemTemplate::PncMapping::PncMapping.pncIdentifier		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMPnc	
BSW Parameter		**BSW Type**
ComMUserPerPnc		EcucReferenceDef
BSW Description		
Reference to the ComMUsers that correspond to this PNC. ImplementationType: COMM_UserHandleType		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet	
BSW Parameter		**BSW Type**
ComMPncEnabled		EcucBooleanParamDef
BSW Description		
Defines whether in this configuration set the partial networking is enabled. true: Enabled false: Disabled		

M2 Template	M2 Description	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context		
ComM	ComM/ComMConfigSet		
BSW Parameter		**BSW Type**	
ComMUser		EcucParamConfContainerDef	
BSW Description			
This container contains a list of identifiers that are needed to refer to a user in the system which is designated to request Communication modes.			
M2 Template	**M2 Description**		
TPS_SWCT, TPS_BSWMDT			
M2 Parameter			
CommonStructure::ServiceNeeds::ComMgrUserNeeds			
Mapping Rule			**Mapping Type**
In case the owner of the ComMgrUserNeeds is a BSW module then the ComMUser.shortName = {capitalizedMip}_{ServiceDependency.symbolicNameProps.symbol}. In case the owner of the ComMgrUserNeeds is a software component then the ComMUser.shortName= {AtomicSwComponentType.shortName}_{ServiceDependency.symbolicNameProps.symbol}.			full

BSW Module	BSW Context		
ComM	ComM/ComMConfigSet/ComMUser		
BSW Parameter		**BSW Type**	
ComMUserEcucPartitionRef		EcucReferenceDef	
BSW Description			
Denotes in which "EcucPartition" the requester is executed. When the partition is stopped, the communication request shall be cancelled in the ComM to avoid a stay-awake situation of the bus due to a stopped partition.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	BSW Context	
ComM	ComM/ComMConfigSet/ComMUser	
BSW Parameter		**BSW Type**
ComMUserIdentifier		EcucIntegerParamDef
BSW Description		

An identifier that is needed to refer to a user in the system which is designated to request Communication Modes. ImplementationType: ComM_UserHandleType		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
ComM	ComM	
BSW Parameter		**BSW Type**
ComMGeneral		EcucParamConfContainerDef
BSW Description		
General configuration parameters of the Communication Manager.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
ComM	ComM/ComMGeneral	
BSW Parameter		**BSW Type**
ComMDevErrorDetect		EcucBooleanParamDef
BSW Description		
Switches the Development Error Detection and Notification ON or OFF. true: Enabled false: Disabled		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
ComM	ComM/ComMGeneral	
BSW Parameter		**BSW Type**
ComMDirectUserMapping		EcucBooleanParamDef
BSW Description		

If this parameter is set to true the configuration tool shall automatically create a ComMUser per ComMPnc and a ComMUser per ComMChannel. The shortName of the generated ComMUsers shall follow the following naming convention: PNCUser_ComMPncId, e.g. PNCUser_13 ChannelUser_ComMChannelId, e.g. ChannelUser_25 Restriction: ComMUser, which are created due to this configuration parameter, shall not be used by SWCs (only available for BswM).	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**
ComM	ComM/ComMGeneral
BSW Parameter	**BSW Type**
ComMEcuGroupClassification	EcucIntegerParamDef
BSW Description	
Defines whether a mode inhibition affects the ECU or not. Examples: 000: No mode inhibition can be activated 001: Wake up inhibition can be enabled Forcing into COMM_NO_COMMUNICATION mode shall be switched on if ComMNmVariant=PASSIVE.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**
ComM	ComM/ComMGeneral
BSW Parameter	**BSW Type**
ComMGlobalNvMBlockDescriptor	EcucSymbolicNameReferenceDef
BSW Description	
Reference to NVRAM block containing the none volatile data. If this parameter is not configured it means that no NVRam is used at all.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	BSW Context	
ComM	ComM/ComMGeneral	
BSW Parameter		**BSW Type**
ComMModeLimitationEnabled		EcucBooleanParamDef
BSW Description		
true if mode limitation functionality shall be enabled. true: Enabled false: Disabled		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
ComM	ComM/ComMGeneral	
BSW Parameter		**BSW Type**
ComMNmPassiveModeEnable		EcucBooleanParamDef
BSW Description		
Enables support of Passive Mode.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
ComM	ComM/ComMGeneral	
BSW Parameter		**BSW Type**
ComMPncGatewayEnabled		EcucBooleanParamDef
BSW Description		
Enables or disables support of Partial Network Gateway. False: Partial Networking Gateway is disabled True: Partial Networking Gateway is enabled		
M2 Template	**M2 Description**	
TPS_SYST	Defines if this EcuInstance shall implement the PncGateway functionality on this CommunicationConnector and its respective PhysicalChannel. Several EcuInstances on the same PhysicalChannel can have the PncGateway functionality enabled, but only one of them shall have the pncGatewayType "active".	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreTopology::CommunicationConnector.pncGatewayType		
Mapping Rule		**Mapping Type**
If in the Ecu Extract the communicationConnector.pncGatewayType is set to active this parameter shall be set to true.		full

BSW Module	BSW Context	
ComM	ComM/ComMGeneral	
BSW Parameter		**BSW Type**

ComMPncPrepareSleepTimer		EcucFloatParamDef	
BSW Description			
Time in seconds the PNC state machine shall wait in PNC_PREPARE_SLEEP.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	**BSW Context**		
ComM	ComM/ComMGeneral		
BSW Parameter		**BSW Type**	
ComMPncSupport		EcucBooleanParamDef	
BSW Description			
Enables or disables support of partial networking. False: Partial Networking is disabled True: Partial Networking is enabled			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	**BSW Context**		
ComM	ComM/ComMGeneral		
BSW Parameter		**BSW Type**	
ComMResetAfterForcingNoComm		EcucBooleanParamDef	
BSW Description			
ComM shall perform a reset after entering "No Communication" mode because of an active mode limitation to "No Communication" mode. true: Enabled false: Disabled			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	**BSW Context**	
ComM	ComM/ComMGeneral	
BSW Parameter		**BSW Type**
ComMSynchronousWakeUp		EcucBooleanParamDef
BSW Description		

Wake up of one channel shall lead to a wake up of all channels if true. true: Enabled false: Disabled		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
ComM	ComM/ComMGeneral	
BSW Parameter		**BSW Type**
ComMTMinFullComModeDuration		EcucFloatParamDef
BSW Description		
Minimum time duration in seconds, spent in the COMM_FULL_COMMUNICATION sub-state COMM_FULL_COM_NETWORK_REQUESTED.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
ComM	ComM/ComMGeneral	
BSW Parameter		**BSW Type**
ComMVersionInfoApi		EcucBooleanParamDef
BSW Description		
Switches the possibility to read the published information with the service ComM_GetPublishedInformation(). true: Enabled false: Disabled		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
ComM	ComM/ComMGeneral	
BSW Parameter		**BSW Type**
ComMWakeupInhibitionEnabled		EcucBooleanParamDef
BSW Description		
true if wake up inhibition functionality enabled. true: Enabled false: Disabled		

M2 Template	M2 Description	
M2 Parameter		
Mapping Rule		**Mapping Type**

D.1.7 Xcp

BSW Module	BSW Context		
Xcp	Xcp		
BSW Parameter		**BSW Type**	
XcpConfig		EcucParamConfContainerDef	
BSW Description			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	BSW Context		
Xcp	Xcp/XcpConfig		
BSW Parameter		**BSW Type**	
XcpDaqList		EcucParamConfContainerDef	
BSW Description			
This container contains the configuration of the DAQs.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	BSW Context		
Xcp	Xcp/XcpConfig/XcpDaqList		
BSW Parameter		**BSW Type**	
XcpDaqListNumber		EcucIntegerParamDef	
BSW Description			
Index number of the DAQ list			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	BSW Context

Xcp	Xcp/XcpConfig/XcpDaqList		
BSW Parameter		**BSW Type**	
XcpDaqListType		EcucEnumerationParamDef	
BSW Description			
This indicates whether this DAQ list represents a DAQ or a STIM.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	**BSW Context**		
Xcp	Xcp/XcpConfig/XcpDaqList		
BSW Parameter		**BSW Type**	
XcpDto		EcucParamConfContainerDef	
BSW Description			
This container collects data transfer object specific parameters for the DAQ list.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	**BSW Context**		
Xcp	Xcp/XcpConfig/XcpDaqList/XcpDto		
BSW Parameter		**BSW Type**	
XcpDto2PduMapping		EcucChoiceReferenceDef	
BSW Description			
This reference specifies the mapping of the DTO to the PDUs from the lower-layer interfaces (CanIf, FrIf, SoAd and Cdd). A reference to a XcpRxPdu is only feasible if the the DaqListType is DAQ_STIM. A reference to a XcpTxPdu is only feasible if the DaqListType is DAQ.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	**BSW Context**	
Xcp	Xcp/XcpConfig/XcpDaqList/XcpDto	
BSW Parameter		**BSW Type**
XcpDtoPid		EcucIntegerParamDef
BSW Description		
Packet identifier (PID) of the DTO that identifies the ODT the content of the DTO.		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**	
Xcp	Xcp/XcpConfig/XcpDaqList	
BSW Parameter		**BSW Type**
XcpMaxOdt		EcucIntegerParamDef
BSW Description		
MAX_ODT indicates the maximum amount of ODTs in this DAQ list (STATIC configuration)		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
Xcp	Xcp/XcpConfig/XcpDaqList	
BSW Parameter		**BSW Type**
XcpMaxOdtEntries		EcucIntegerParamDef
BSW Description		
This parameter indicates the maximum amount of entries in an ODT of this DAQ list (STATIC configuration).		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
Xcp	Xcp/XcpConfig/XcpDaqList	
BSW Parameter		**BSW Type**
XcpOdt		EcucParamConfContainerDef
BSW Description		
This container contains ODT-specific parameter for the DAQ list.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
Xcp	Xcp/XcpConfig/XcpDaqList/XcpOdt	
BSW Parameter		**BSW Type**

XcpOdt2DtoMapping		EcucReferenceDef	
BSW Description			
This reference maps the ODT to the according DTO in which it will be transmitted.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	**BSW Context**		
Xcp	Xcp/XcpConfig/XcpDaqList/XcpOdt		
BSW Parameter		**BSW Type**	
XcpOdtEntry		EcucParamConfContainerDef	
BSW Description			
This container collects all configuration parameters that comprise an ODT entry.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	**BSW Context**		
Xcp	Xcp/XcpConfig/XcpDaqList/XcpOdt/XcpOdtEntry		
BSW Parameter		**BSW Type**	
XcpOdtEntryAddress		EcucLinkerSymbolDef	
BSW Description			
Memory address that the ODT entry is referencing to.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	**BSW Context**		
Xcp	Xcp/XcpConfig/XcpDaqList/XcpOdt/XcpOdtEntry		
BSW Parameter		**BSW Type**	
XcpOdtEntryBitOffset		EcucIntegerParamDef	
BSW Description			
Represent the bit offset in case of the element represents status bit.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	BSW Context	
Xcp	Xcp/XcpConfig/XcpDaqList/XcpOdt/XcpOdtEntry	
BSW Parameter		**BSW Type**
XcpOdtEntryLength		EcucIntegerParamDef
BSW Description		
Length of the referenced memory area that is referenced by the ODT entry.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
Xcp	Xcp/XcpConfig/XcpDaqList/XcpOdt/XcpOdtEntry	
BSW Parameter		**BSW Type**
XcpOdtEntryNumber		EcucIntegerParamDef
BSW Description		
Index number of the ODT entry		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
Xcp	Xcp/XcpConfig/XcpDaqList/XcpOdt	
BSW Parameter		**BSW Type**
XcpOdtEntryMaxSize		EcucIntegerParamDef
BSW Description		
This parameter indicates the upper limit for the size of the element described by an ODT entry. Depending on the DaqListType this ODT belongs to it describes the limit for a DAQ (MAX_ODT_ENTRY_SIZE_DAQ) or a STIM (MAX_ODT_ENTRY_SIZE_STIM).		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
Xcp	Xcp/XcpConfig/XcpDaqList/XcpOdt	
BSW Parameter		**BSW Type**
XcpOdtNumber		EcucIntegerParamDef
BSW Description		
Index number of this ODT within the DAQ list.		
M2 Template	**M2 Description**	
M2 Parameter		

Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**	
Xcp	Xcp/XcpConfig	
BSW Parameter		**BSW Type**
XcpEventChannel		EcucParamConfContainerDef
BSW Description		
This container contains the configuration of event channels on the XCP slave.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
Xcp	Xcp/XcpConfig/XcpEventChannel	
BSW Parameter		**BSW Type**
XcpEventChannelConsistency		EcucEnumerationParamDef
BSW Description		
Type of consistency used by event channel		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
Xcp	Xcp/XcpConfig/XcpEventChannel	
BSW Parameter		**BSW Type**
XcpEventChannelMaxDaqList		EcucIntegerParamDef
BSW Description		
Maximum amount of DAQ lists that are handled by this event channel.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
Xcp	Xcp/XcpConfig/XcpEventChannel	
BSW Parameter		**BSW Type**
XcpEventChannelNumber		EcucIntegerParamDef
BSW Description		
Index number of the event channel.		

M2 Template	M2 Description	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
Xcp	Xcp/XcpConfig/XcpEventChannel	
BSW Parameter		**BSW Type**
XcpEventChannelPriority		EcucIntegerParamDef
BSW Description		
Priority of the event channel		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
Xcp	Xcp/XcpConfig/XcpEventChannel	
BSW Parameter		**BSW Type**
XcpEventChannelTimeCycle		EcucIntegerParamDef
BSW Description		
The event channel time cycle indicates which sampling period is used to process this event channel. A value of 0 means 'Not cyclic'.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
Xcp	Xcp/XcpConfig/XcpEventChannel	
BSW Parameter		**BSW Type**
XcpEventChannelTimeUnit		EcucEnumerationParamDef
BSW Description		
This configuration parameter indicates the unit of the event channel time cycle.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context
Xcp	Xcp/XcpConfig/XcpEventChannel

<table>
<tr><th colspan="2">BSW Parameter</th><th colspan="2">BSW Type</th></tr>
<tr><td colspan="2">XcpEventChannelTriggeredDaqListRef</td><td colspan="2">EcucReferenceDef</td></tr>
<tr><th colspan="4">BSW Description</th></tr>
<tr><td colspan="4">References all DAQ lists that are trigged by this event channel.</td></tr>
<tr><th>M2 Template</th><th colspan="3">M2 Description</th></tr>
<tr><td></td><td colspan="3"></td></tr>
<tr><th colspan="4">M2 Parameter</th></tr>
<tr><td colspan="4"></td></tr>
<tr><th colspan="3">Mapping Rule</th><th>Mapping Type</th></tr>
<tr><td colspan="3"></td><td></td></tr>
</table>

<table>
<tr><th>BSW Module</th><th colspan="3">BSW Context</th></tr>
<tr><td>Xcp</td><td colspan="3">Xcp/XcpConfig/XcpEventChannel</td></tr>
<tr><th colspan="2">BSW Parameter</th><th colspan="2">BSW Type</th></tr>
<tr><td colspan="2">XcpEventChannelType</td><td colspan="2">EcucEnumerationParamDef</td></tr>
<tr><th colspan="4">BSW Description</th></tr>
<tr><td colspan="4">This configuration parameter indicates what kind of DAQ list can be allocated to this event channel.</td></tr>
<tr><th>M2 Template</th><th colspan="3">M2 Description</th></tr>
<tr><td></td><td colspan="3"></td></tr>
<tr><th colspan="4">M2 Parameter</th></tr>
<tr><td colspan="4"></td></tr>
<tr><th colspan="3">Mapping Rule</th><th>Mapping Type</th></tr>
<tr><td colspan="3"></td><td></td></tr>
</table>

<table>
<tr><th>BSW Module</th><th colspan="3">BSW Context</th></tr>
<tr><td>Xcp</td><td colspan="3">Xcp/XcpConfig</td></tr>
<tr><th colspan="2">BSW Parameter</th><th colspan="2">BSW Type</th></tr>
<tr><td colspan="2">XcpPdu</td><td colspan="2">EcucChoiceContainerDef</td></tr>
<tr><th colspan="4">BSW Description</th></tr>
<tr><td colspan="4">Contains PDU information. A PDU may be either a transmission PDU or a reception PDU.</td></tr>
<tr><th>M2 Template</th><th colspan="3">M2 Description</th></tr>
<tr><td>TPS_SYST</td><td colspan="3">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.</td></tr>
<tr><th colspan="4">M2 Parameter</th></tr>
<tr><td colspan="4">SystemTemplate::Fibex::FibexCore::CoreCommunication::GeneralPurposeIPdu</td></tr>
<tr><th colspan="3">Mapping Rule</th><th>Mapping Type</th></tr>
<tr><td colspan="3">Create this container if a GeneralPurposeIPdu with the category "Xcp" is defined in the Ecu Extract.</td><td>full</td></tr>
</table>

<table>
<tr><th>BSW Module</th><th colspan="3">BSW Context</th></tr>
<tr><td>Xcp</td><td colspan="3">Xcp/XcpConfig/XcpPdu</td></tr>
<tr><th colspan="2">BSW Parameter</th><th colspan="2">BSW Type</th></tr>
<tr><td colspan="2">XcpRxPdu</td><td colspan="2">EcucParamConfContainerDef</td></tr>
<tr><th colspan="4">BSW Description</th></tr>
<tr><td colspan="4">This container specifies received PDUs.</td></tr>
<tr><th>M2 Template</th><th colspan="3">M2 Description</th></tr>
<tr><td>TPS_SYST</td><td colspan="3">References to the IPduPort on every ECU of the system which sends and/or receives the I-PDU.</td></tr>
<tr><th colspan="4">M2 Parameter</th></tr>
</table>

SystemTemplate::Fibex::FibexCore::CoreCommunication::PduTriggering.iPduPort	
Mapping Rule	**Mapping Type**
The direction of the Pdu can be derived from the PduTriggering that refers to the GeneralPurposePdu that represents the XcpPdu.	full

BSW Module	**BSW Context**	
Xcp	Xcp/XcpConfig/XcpPdu/XcpRxPdu	
BSW Parameter		**BSW Type**
XcpRxPduId		EcucIntegerParamDef
BSW Description		
ID of the PDU that will be received via a Xcp_<module>RxIndication.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
Xcp	Xcp/XcpConfig/XcpPdu/XcpRxPdu	
BSW Parameter		**BSW Type**
XcpRxPduRef		EcucReferenceDef
BSW Description		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
Xcp	Xcp/XcpConfig/XcpPdu	
BSW Parameter		**BSW Type**
XcpTxPdu		EcucParamConfContainerDef
BSW Description		
This container specifies transmission PDUs.		
M2 Template	**M2 Description**	
TPS_SYST	References to the IPduPort on every ECU of the system which sends and/or receives the I-PDU.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::PduTriggering.iPduPort		
Mapping Rule		**Mapping Type**
The direction of the Pdu can be derived from the PduTriggering that refers to the GeneralPurposePdu that represents the XcpPdu.		full

BSW Module	**BSW Context**	
Xcp	Xcp/XcpConfig/XcpPdu/XcpTxPdu	
BSW Parameter		**BSW Type**

XcpTxPduId	EcucIntegerParamDef
BSW Description	
The PDU identifier, which has to be used by the lower layer BSW module for TxConfirmations or TriggerTransmits.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
Xcp	Xcp/XcpConfig/XcpPdu/XcpTxPdu
BSW Parameter	**BSW Type**
XcpTxPduRef	EcucReferenceDef
BSW Description	
Reference to the external PDU definition.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
Xcp	Xcp
BSW Parameter	**BSW Type**
XcpGeneral	EcucParamConfContainerDef
BSW Description	
This container contains the general configuration parameters of the XCP.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**
Xcp	Xcp/XcpGeneral
BSW Parameter	**BSW Type**
XcpCounterRef	EcucReferenceDef
BSW Description	
This parameter contains a reference to the counter, which is used by XCP.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	BSW Context	
Xcp	Xcp/XcpGeneral	
BSW Parameter		**BSW Type**
XcpDaqConfigType		EcucEnumerationParamDef
BSW Description		
Sets the DAQ_CONFIG_TYPE bit within the DAQ_PROPERTIES parameter to "static" or to "dynamic". If DAQ_STATIC is selected, the DAQ_CONFIG_TYPE bit is set to "0". If DAQ_DYNAMIC is selected, the DAQ_CONFIG_TYPE bit is set to "1".		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
Xcp	Xcp/XcpGeneral	
BSW Parameter		**BSW Type**
XcpDaqCount		EcucIntegerParamDef
BSW Description		
Indicates the number of DAQ lists for dynamic configuration.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
Xcp	Xcp/XcpGeneral	
BSW Parameter		**BSW Type**
XcpDevErrorDetect		EcucBooleanParamDef
BSW Description		
Switches the Development Error Detection and Notification on or off. TRUE: Development Error Detection and Notification on FALSE: Development Error Detection and Notification off		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
Xcp	Xcp/XcpGeneral	
BSW Parameter		**BSW Type**
XcpIdentificationFieldType		EcucEnumerationParamDef
BSW Description		
Type of Identification Field the slave will use when transferring DAQ Packets to the master. The master has to use the same Type of Identification Field when transferring STIM Packets to the slave.		

M2 Template	M2 Description	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
Xcp	Xcp/XcpGeneral	
BSW Parameter		**BSW Type**
XcpMainFunctionPeriod		EcucFloatParamDef
BSW Description		
The XCP does not require this information but the BSW scheduler, which invokes the main function, needs it in order to plan its tasks.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
Xcp	Xcp/XcpGeneral	
BSW Parameter		**BSW Type**
XcpMaxCto		EcucIntegerParamDef
BSW Description		
MAX_CTO shows the maximum length of a CTO packet in bytes.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
Xcp	Xcp/XcpGeneral	
BSW Parameter		**BSW Type**
XcpMaxDto		EcucIntegerParamDef
BSW Description		
MAX_DTO shows the maximum length of a DTO packet in bytes.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context
Xcp	Xcp/XcpGeneral

BSW Parameter		BSW Type	
XcpMaxEventChannel		EcucIntegerParamDef	
BSW Description			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	BSW Context		
Xcp	Xcp/XcpGeneral		
BSW Parameter		**BSW Type**	
XcpMinDaq		EcucIntegerParamDef	
BSW Description			
Indicates the number of predefined, read only DAQ lists on the XCP slave.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	BSW Context		
Xcp	Xcp/XcpGeneral		
BSW Parameter		**BSW Type**	
XcpOdtCount		EcucIntegerParamDef	
BSW Description			
This parameter indicates the amount of ODTs of a DAQ list using dynamic DAQ list configuration.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	BSW Context		
Xcp	Xcp/XcpGeneral		
BSW Parameter		**BSW Type**	
XcpOdtEntriesCount		EcucIntegerParamDef	
BSW Description			
Indicates the amount of entries into an ODT using dynamic DAQ list configuration.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	BSW Context
Xcp	Xcp/XcpGeneral
BSW Parameter	**BSW Type**
XcpOdtEntrySizeDaq	EcucIntegerParamDef
BSW Description	
Indicates the size of an element described by an ODT entry to the DaqListType for a DAQ.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	BSW Context
Xcp	Xcp/XcpGeneral
BSW Parameter	**BSW Type**
XcpOdtEntrySizeStim	EcucIntegerParamDef
BSW Description	
Indicates the size of an element described by an ODT entry to the DaqListType for a stim.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	BSW Context
Xcp	Xcp/XcpGeneral
BSW Parameter	**BSW Type**
XcpOnCanEnabled	EcucBooleanParamDef
BSW Description	
Enabling of XCPonCAN functionality	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	BSW Context
Xcp	Xcp/XcpGeneral
BSW Parameter	**BSW Type**
XcpOnCddEnabled	EcucBooleanParamDef
BSW Description	
Enabling of XCPonCdd functionality	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	BSW Context	
Xcp	Xcp/XcpGeneral	
BSW Parameter		**BSW Type**
XcpOnEthernetEnabled		EcucBooleanParamDef
BSW Description		
Enabling of XCPonEthernet functionality		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
Xcp	Xcp/XcpGeneral	
BSW Parameter		**BSW Type**
XcpOnFlexRayEnabled		EcucBooleanParamDef
BSW Description		
Enabling of XCPonFlexRay functionality		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
Xcp	Xcp/XcpGeneral	
BSW Parameter		**BSW Type**
XcpPrescalerSupported		EcucBooleanParamDef
BSW Description		
This parameter enables and disables the support for Prescaler support. True is Enabled, False is disabled		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
Xcp	Xcp/XcpGeneral	
BSW Parameter		**BSW Type**
XcpSuppressTxSupport		EcucBooleanParamDef
BSW Description		

Switches the support of suppressing transmission of PDUs per communication channel on or off. TRUE: Suppressing of TxPDUs supported FALSE: Suppressing of TxPDUs not supported		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
Xcp	Xcp/XcpGeneral	
BSW Parameter		**BSW Type**
XcpTimestampTicks		EcucIntegerParamDef
BSW Description		
This parameter defines the timestamp that will increment based TIMESTAMP_TICKS per unit and wrap around if an overflow occurs.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
Xcp	Xcp/XcpGeneral	
BSW Parameter		**BSW Type**
XcpTimestampType		EcucEnumerationParamDef
BSW Description		
This parameter indicates the number of bytes used for the timestamp field. In case No_TIME_STAMP is selected the timestamp field is not available.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
Xcp	Xcp/XcpGeneral	
BSW Parameter		**BSW Type**
XcpTimestampUnit		EcucEnumerationParamDef
BSW Description		
This parameter indicates the resolution of the data acquisition clock of the slave when transferring data to master.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context		
Xcp	Xcp/XcpGeneral		
BSW Parameter		**BSW Type**	
XcpVersionInfoApi		EcucBooleanParamDef	
BSW Description			
Enables/disables the existence of the XCP_GetVersionInfo() API service. TRUE: XCP_GetVersionInfo() API service exists FALSE: XCP_GetVersionInfo() API service does not exist			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

D.2 Can

D.2.1 Can Driver Mapping

BSW Module	BSW Context		
Can	Can		
BSW Parameter		**BSW Type**	
CanConfigSet		EcucParamConfContainerDef	
BSW Description			
This is the multiple configuration set container for CAN Driver			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
Can	Can/CanConfigSet		
BSW Parameter		**BSW Type**	
CanController		EcucParamConfContainerDef	
BSW Description			
This container contains the configuration parameters of the CAN controller(s).			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController	
BSW Parameter		**BSW Type**

CanBusoffProcessing		EcucEnumerationParamDef	
BSW Description			
Enables / disables API Can_MainFunction_BusOff() for handling busoff events in polling mode.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
Can	Can/CanConfigSet/CanController		
BSW Parameter		**BSW Type**	
CanControllerActivation		EcucBooleanParamDef	
BSW Description			
Defines if a CAN controller is used in the configuration.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
Can	Can/CanConfigSet/CanController		
BSW Parameter		**BSW Type**	
CanControllerBaseAddress		EcucIntegerParamDef	
BSW Description			
Specifies the CAN controller base address.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
Can	Can/CanConfigSet/CanController		
BSW Parameter		**BSW Type**	
CanControllerBaudrateConfig		EcucParamConfContainerDef	
BSW Description			
This container contains bit timing related configuration parameters of the CAN controller(s).			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**	
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig	
BSW Parameter		**BSW Type**
CanControllerBaudRate		EcucIntegerParamDef
BSW Description		
Specifies the baudrate of the controller in kbps.		
M2 Template	**M2 Description**	
TPS_SYST	channels speed in bits per second	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreTopology::CommunicationCluster.baudrate		
Mapping Rule		**Mapping Type**
SystemTemplate speed is in bps, so divide it by 1000 to get kbps		full

BSW Module	**BSW Context**	
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig	
BSW Parameter		**BSW Type**
CanControllerBaudRateConfigID		EcucIntegerParamDef
BSW Description		
Uniquely identifies a specific baud rate configuration. This ID is used by SetBaudrate API.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig	
BSW Parameter		**BSW Type**
CanControllerFdBaudrateConfig		EcucParamConfContainerDef
BSW Description		
This optional container contains bit timing related configuration parameters of the CAN controller(s) for payload and CRC of a CAN FD frame. If this container exists the controller supports CAN FD frames.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig/CanControllerFdBaudrateConfig	
BSW Parameter		**BSW Type**
CanControllerFdBaudRate		EcucIntegerParamDef
BSW Description		
Specifies the data segment baud rate of the controller in kbps.		
M2 Template	**M2 Description**	
TPS_SYST	Specifies the data segment baud rate of the controller in bits/s.	

M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanTopology::AbstractCanCluster.canFdBaudrate	
Mapping Rule	**Mapping Type**
SystemTemplate speed is in bps, so divide it by 1000 to get kbps	full

BSW Module	BSW Context
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig/CanControllerFdBaudrateConfig
BSW Parameter	**BSW Type**
CanControllerPropSeg	EcucIntegerParamDef
BSW Description	
Specifies propagation delay in time quantas.	
M2 Template	**M2 Description**
TPS_SYST	Specifies propagation delay in time quantas.
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanControllerFdConfiguration.propSeg	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	BSW Context
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig/CanControllerFdBaudrateConfig
BSW Parameter	**BSW Type**
CanControllerSeg1	EcucIntegerParamDef
BSW Description	
Specifies phase segment 1 in time quantas.	
M2 Template	**M2 Description**
TPS_SYST	Specifies phase segment 1 in time quantas. timeSeg1 = Phase_Seg1
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanControllerFdConfiguration.timeSeg1	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	BSW Context
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig/CanControllerFdBaudrateConfig
BSW Parameter	**BSW Type**
CanControllerSeg2	EcucIntegerParamDef
BSW Description	
Specifies phase segment 2 in time quantas.	
M2 Template	**M2 Description**
TPS_SYST	Specifies phase segment 2 in time quantas. timeSeg2 = Phase_Seg2
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanControllerFdConfiguration.timeSeg2	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig/CanControllerFdBaudrateConfig	
BSW Parameter		**BSW Type**
CanControllerSyncJumpWidth		EcucIntegerParamDef
BSW Description		
Specifies the synchronization jump width for the controller in time quantas.		
M2 Template	**M2 Description**	
TPS_SYST	The number of quanta in the Synchronization Jump Width, SJW.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanControllerFdConfiguration.syncJumpWidth		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig/CanControllerFdBaudrateConfig	
BSW Parameter		**BSW Type**
CanControllerTrcvDelayCompensationOffset		EcucIntegerParamDef
BSW Description		
Specifies the Transceiver Delay Compensation Offset in ns. If not specified Transceiver Delay Compensation is disabled.		
M2 Template	**M2 Description**	
TPS_SYST	Specifies the Transceiver Delay Compensation Offset in seconds. If not specified Transceiver Delay Compensation is disabled.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanControllerFdConfiguration.trcvDelayCompensationOffset		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig/CanControllerFdBaudrateConfig	
BSW Parameter		**BSW Type**
CanControllerTxBitRateSwitch		EcucBooleanParamDef
BSW Description		
Specifies if the bit rate switching shall be used for transmissions. If FALSE: CAN FD frames shall be sent without bit rate switching.		
M2 Template	**M2 Description**	
TPS_SYST	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.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanControllerFdConfiguration.txBitRateSwitch		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig	
BSW Parameter		**BSW Type**

CanControllerPropSeg		EcucIntegerParamDef	
BSW Description			
Specifies propagation delay in time quantas.			
M2 Template	**M2 Description**		
TPS_SYST	Specifies propagation delay in time quantas.		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanControllerConfiguration.propSeg			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	**BSW Context**		
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig		
BSW Parameter		**BSW Type**	
CanControllerSeg1		EcucIntegerParamDef	
BSW Description			
Specifies phase segment 1 in time quantas.			
M2 Template	**M2 Description**		
TPS_SYST	Specifies phase segment 1 in time quantas. timeSeg1 = Phase_Seg1		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanControllerConfiguration.timeSeg1			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	**BSW Context**		
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig		
BSW Parameter		**BSW Type**	
CanControllerSeg2		EcucIntegerParamDef	
BSW Description			
Specifies phase segment 2 in time quantas.			
M2 Template	**M2 Description**		
TPS_SYST	Specifies phase segment 2 in time quantas. timeSeg2 = Phase_Seg2		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanControllerConfiguration.timeSeg2			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	**BSW Context**		
Can	Can/CanConfigSet/CanController/CanControllerBaudrateConfig		
BSW Parameter		**BSW Type**	
CanControllerSyncJumpWidth		EcucIntegerParamDef	
BSW Description			
Specifies the synchronization jump width for the controller in time quantas.			
M2 Template	**M2 Description**		
TPS_SYST	The number of quanta in the Synchronization Jump Width, SJW.		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanControllerConfiguration.syncJumpWidth			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController	
BSW Parameter		**BSW Type**
CanControllerDefaultBaudrate		EcucReferenceDef
BSW Description		
Reference to baudrate configuration container configured for the Can Controller.		
M2 Template	**M2 Description**	
TPS_SYST	channels speed in bits per second	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreTopology::CommunicationCluster.speed		
Mapping Rule		**Mapping Type**
SystemTemplate speed is in bps, so divide it by 1000 to get kbps		full

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController	
BSW Parameter		**BSW Type**
CanControllerId		EcucIntegerParamDef
BSW Description		
This parameter provides the controller ID which is unique in a given CAN Driver. The value for this parameter starts with 0 and continue without any gaps.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController	
BSW Parameter		**BSW Type**
CanCpuClockRef		EcucReferenceDef
BSW Description		
Reference to the CPU clock configuration, which is set in the MCU driver configuration		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController	
BSW Parameter		**BSW Type**
CanFilterMask		EcucParamConfContainerDef
BSW Description		
This container contains the configuration (parameters) of the CAN Filter Mask(s). This container is set to obsolete. It is replaced by CanHwFilterMask		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController/CanFilterMask	
BSW Parameter		**BSW Type**
CanFilterMaskValue		EcucIntegerParamDef
BSW Description		
Describes a mask for hardware-based filtering of CAN identifiers. The CAN identifiers of incoming messages are masked with the appropriate CanFilterMaskValue. Bits holding a 0 mean don't care, i.e. do not compare the message's identifier in the respective bit position. The mask shall be build by filling with leading 0. In case of CanIdType EXTENDED or MIXED a 29 bit mask shall be build. In case of CanIdType STANDARD a 11 bit mask shall be build		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController	
BSW Parameter		**BSW Type**
CanRxProcessing		EcucEnumerationParamDef
BSW Description		
Enables / disables API Can_MainFunction_Read() for handling PDU reception events in polling mode.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController	
BSW Parameter		**BSW Type**
CanTTController		EcucParamConfContainerDef
BSW Description		
CanTTController is specified in the SWS TTCAN and contains the configuration parameters of the TTCAN controller(s) (which are needed in addition to the configuration parameters of the CAN controller(s)). This container is only included and valid if TTCAN is supported by the controller, enabled (see CanSupportTTCANRef, ECUC_Can_00430), and used.		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**	
Can	Can/CanConfigSet/CanController/CanTTController	
BSW Parameter		**BSW Type**
CanTTControllerApplWatchdogLimit		EcucIntegerParamDef
BSW Description		
Defines the maximum time period (unit is 256 times NTU) after which the application has to serve the watchdog.		
M2 Template	**M2 Description**	
TPS_SYST	The Appl_Watchdog_Limit shall be an 8-bit value specifying the period for the application watchdog in Appl_Watchdog_Limit times 256 NTUs.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ttcan::TtcanTopology::TtcanCommunicationController.applWatch-dogLimit		
Mapping Rule		**Mapping Type**
1:1 mapping		local

BSW Module	**BSW Context**	
Can	Can/CanConfigSet/CanController/CanTTController	
BSW Parameter		**BSW Type**
CanTTControllerCycleCountMax		EcucIntegerParamDef
BSW Description		
Defines the value for cycle_count_max. Allowed values: 0x00: 1 basic cycle 0x01: 2 basic cycles 0x03: 4 basic cycles 0x07: 8 basic cycles 0x0F: 16 basic cycles 0x1F: 32 basic cycles 0x3F: 64 basic cycles		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
Can	Can/CanConfigSet/CanController/CanTTController	
BSW Parameter		**BSW Type**
CanTTControllerExpectedTxTrigger		EcucIntegerParamDef
BSW Description		
Number of expected_tx_trigger.		
M2 Template	**M2 Description**	
TPS_SYST	The Expected_Tx_Trigger shall be an eight (8) bit value which limits the number of messages the FSE may try to transmit in one matrix cycle.	

M2 Parameter	
SystemTemplate::Fibex::Fibex4Ttcan::TtcanTopology::TtcanCommunicationController.expectedTxTrigger	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController/CanTTController	
BSW Parameter		**BSW Type**
CanTTControllerExternalClockSynchronisation		EcucBooleanParamDef
BSW Description		
Enables/disables the external clock synchronization. TRUE: External clock synchronization enabled. FALSE: External clock synchronization disabled. This parameter shall only be configurable if parameter CanTTControllerLevel2 equals TRUE.		
M2 Template	**M2 Description**	
TPS_SYST	One bit shall be used to configure whether or not external clock synchronisation will be allowed during runtime (only Level 2).	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ttcan::TtcanTopology::TtcanCommunicationController.externalClockSynchronisation		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController/CanTTController	
BSW Parameter		**BSW Type**
CanTTControllerGlobalTimeFiltering		EcucBooleanParamDef
BSW Description		
Enables/disables the global time filtering. TRUE: Global time filtering enabled. FALSE: Global time filtering disabled. This parameter shall only be configurable if parameter CanTTControllerLevel2 equals TRUE.		
M2 Template	**M2 Description**	
	Enables/disables global time filtering	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController/CanTTController	
BSW Parameter		**BSW Type**
CanTTControllerInitialRefOffset		EcucIntegerParamDef
BSW Description		

Defines the initial value for ref trigger offset.		
M2 Template	**M2 Description**	
TPS_SYST	The Initial_Ref_Offset shall be an eight (8) bit value for the initialisation of Ref_Trigger_Offset.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ttcan::TtcanTopology::TtcanCommunicationController.initialRefOffset		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Can	Can/CanConfigSet/CanController/CanTTController	
BSW Parameter		**BSW Type**
CanTTControllerInterruptEnable		EcucIntegerParamDef
BSW Description		
Enables/disables the respective interrupts. Bit Position set to 1: Enable respective interrupt. Bit Position set to 0: Disable respective interrupt. Bit Position / Interrupt Source: 10: Application Watchdog. 9: Watch Trigger reached. 8: Initialization Watch Trigger reached. 7: Change of Error Level. 6: Tx Overflow. 5: Tx Underflow. 4: Global Time Error. 3: Gap. 2: Start of Cycle. 1: Time Discontinuity. 0: Master State Change. Bit position "1: Time Discontinuity" and "4: Global Time Error" shall only be configurable if parameter CanTTControllerLevel2 equals TRUE.		
M2 Template	**M2 Description**	
	Enables/disables interrupts	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
Can	Can/CanConfigSet/CanController/CanTTController	
BSW Parameter		**BSW Type**
CanTTControllerLevel2		EcucBooleanParamDef
BSW Description		
Defines whether Level 2 or Level 1 is used. TRUE: Level 2. FALSE: Level 1. If this parameter is set to FALSE then all parameters with dependency to CanTTControllerLevel2 need not be configured.		

M2 Template	M2 Description	
TPS_SYST	One bit shall be used to distinguish between Level 1 and Level 2.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ttcan::TtcanTopology::TtcanCommunicationController.timeTriggeredCanLevel		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController/CanTTController	
BSW Parameter		**BSW Type**
CanTTControllerNTUConfig		EcucFloatParamDef
BSW Description		
Defines the config value for NTU (network time unit). Value given in microseconds. The value configured shall be greater than 0. Together with the local oscillator period, the TUR (time unit ratio) can be derived from the NTU. This parameter shall only be configurable if parameter CanTTControllerLevel2 equals TRUE.		
M2 Template	**M2 Description**	
TPS_SYST	Unit measuring all times and providing a constant of the whole network. For level 1, this is always the CAN bit time. Unit: seconds.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ttcan::TtcanTopology::TtcanCluster.ntu		
Mapping Rule		**Mapping Type**
NTU = system clock period x (TUR Numerator / TUR Denominator)		full

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController/CanTTController	
BSW Parameter		**BSW Type**
CanTTControllerOperationMode		EcucEnumerationParamDef
BSW Description		
Defines the operation mode.		
M2 Template	**M2 Description**	
TPS_SYST	Operation mode: Time-triggered or event synchronized time-triggered	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ttcan::TtcanTopology::TtcanCluster.operationMode		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController/CanTTController	
BSW Parameter		**BSW Type**
CanTTControllerSyncDeviation		EcucFloatParamDef
BSW Description		
Defines the maximum synchronization deviation: Given as a percentage value of the NTU (network time unit). The value configured shall be greater than 0. This parameter shall only be configurable if parameter CanTTControllerLevel2 equals TRUE.		
M2 Template	**M2 Description**	
	Defines maximum tolerated synchronization deviation	
M2 Parameter		

Mapping Rule	**Mapping Type**
Synchronisation Deviation <= 2ˆ (CanTTSyncDeviation + 5).	local

BSW Module	**BSW Context**	
Can	Can/CanConfigSet/CanController/CanTTController	
BSW Parameter		**BSW Type**
CanTTControllerTURRestore		EcucBooleanParamDef
BSW Description		
Enables/disables the TUR restore. Note that the value configured for TUR can be derived from the value configured for NTU and the local oscillator preriod. TRUE: TUR restore enabled. FALSE: TUR restore disabled. This parameter shall only be configurable if parameter CanTTControllerLevel2 equals TRUE.		
M2 Template	**M2 Description**	
	Enables/disables TUR restore	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
Can	Can/CanConfigSet/CanController/CanTTController	
BSW Parameter		**BSW Type**
CanTTControllerTimeMaster		EcucBooleanParamDef
BSW Description		
Defines whether the controller acts as a potential time master. TRUE: Potential time master. FALSE: Time slave.		
M2 Template	**M2 Description**	
TPS_SYST	Master-salve mode: Potential time master or slave	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ttcan::TtcanTopology::TtcanCommunicationController.master		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Can	Can/CanConfigSet/CanController/CanTTController	
BSW Parameter		**BSW Type**
CanTTControllerTimeMasterPriority		EcucIntegerParamDef
BSW Description		
Defines the time master priority.		
M2 Template	**M2 Description**	
TPS_SYST	The time master priority shall contain a three bit value for the priority of the current time master (the last three bits of the identifier of the reference message). This can be derived from the frame-triggering's triggers.	
M2 Parameter		

SystemTemplate::Fibex::Fibex4Ttcan::TtcanTopology::TtcanCommunicationController.timeMasterPriority	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**	
Can	Can/CanConfigSet/CanController/CanTTController	
BSW Parameter		**BSW Type**
CanTTControllerTxEnableWindowLength		EcucIntegerParamDef
BSW Description		
Length of the tx enable window given in CAN bit times. Definition parameter "CanTTControllerTxEnableWindowlength" is used such that: Length of enable window = CanTTControllerTxEnableWindowLength + 1		
M2 Template	**M2 Description**	
TPS_SYST	Length of the tx enable window given in CAN bit times	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ttcan::TtcanTopology::TtcanCommunicationController.txEnableWindowLength		
Mapping Rule		**Mapping Type**
Length of enable window = CanTTControllerTxEnableWindowLength + 1		full

BSW Module	**BSW Context**	
Can	Can/CanConfigSet/CanController/CanTTController	
BSW Parameter		**BSW Type**
CanTTControllerWatchTriggerGapTimeMark		EcucIntegerParamDef
BSW Description		
watch trigger time mark after a gap		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
Can	Can/CanConfigSet/CanController/CanTTController	
BSW Parameter		**BSW Type**
CanTTControllerWatchTriggerTimeMark		EcucIntegerParamDef
BSW Description		
watch trigger time mark		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**
Can	Can/CanConfigSet/CanController/CanTTController

BSW Parameter			BSW Type	
CanTTIRQProcessing			EcucEnumerationParamDef	
BSW Description				
Enables / disables API Can_MainFunction_BusOff() for handling busoff events in polling mode.				
M2 Template	**M2 Description**			
M2 Parameter				
Mapping Rule				**Mapping Type**
				local

BSW Module	BSW Context			
Can	Can/CanConfigSet/CanController			
BSW Parameter			**BSW Type**	
CanTxProcessing			EcucEnumerationParamDef	
BSW Description				
Enables / disables API Can_MainFunction_Write() for handling PDU transmission events in polling mode.				
M2 Template	**M2 Description**			
M2 Parameter				
Mapping Rule				**Mapping Type**
				local

BSW Module	BSW Context			
Can	Can/CanConfigSet/CanController			
BSW Parameter			**BSW Type**	
CanWakeupFunctionalityAPI			EcucBooleanParamDef	
BSW Description				
Adds / removes the service Can_CheckWakeup() from the code. True: Can_CheckWakeup can be used. False: Can_CheckWakeup cannot be used.				
M2 Template	**M2 Description**			
M2 Parameter				
Mapping Rule				**Mapping Type**

BSW Module	BSW Context		
Can	Can/CanConfigSet/CanController		
BSW Parameter			**BSW Type**
CanWakeupProcessing			EcucEnumerationParamDef
BSW Description			
Enables / disables API Can_MainFunction_Wakeup() for handling wakeup events in polling mode.			
M2 Template	**M2 Description**		
M2 Parameter			

Mapping Rule	Mapping Type
	local

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController	
BSW Parameter		**BSW Type**
CanWakeupSourceRef		EcucSymbolicNameReferenceDef
BSW Description		
This parameter contains a reference to the Wakeup Source for this controller as defined in the ECU State Manager. Implementation Type: reference to EcuM_WakeupSourceType		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanController	
BSW Parameter		**BSW Type**
CanWakeupSupport		EcucBooleanParamDef
BSW Description		
CAN driver support for wakeup over CAN Bus.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Can	Can/CanConfigSet	
BSW Parameter		**BSW Type**
CanHardwareObject		EcucParamConfContainerDef
BSW Description		
This container contains the configuration (parameters) of CAN Hardware Objects.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanHardwareObject	
BSW Parameter		**BSW Type**
CanControllerRef		EcucReferenceDef

BSW Description		
Reference to CAN Controller to which the HOH is associated to.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanHardwareObject	
BSW Parameter		**BSW Type**
CanFilterMaskRef		EcucReferenceDef
BSW Description		
Reference to the filter mask that is used for hardware filtering together with the CAN_ID_VALUE. Different CanHardwareObjects with different CanIdTypes (STANDARD, MIXED, EXTENDED) can share the same CanFilterMask (i.e., the CanFilterMaskRef parameters of these CanHardwareObjects reference the very same CanFilterMask container). This shall be allowed and must be supported by the configuration generators. The CanFilterMaskRef is omitted for 1) CanHardwareObjects with CanObjectType set to TRANSMIT 2) CanHardwareObjects with CanObjectType set to RECEIVE if only a single Can ID shall be received via this CanHardwareObjects (i.e., exact match with CanIdValue) This parameter is set to obsolete and is replaced by CanHwFilterMask		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanHardwareObject	
BSW Parameter		**BSW Type**
CanHandleType		EcucEnumerationParamDef
BSW Description		
Specifies the type (Full-CAN or Basic-CAN) of a hardware object.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanHardwareObject	
BSW Parameter		**BSW Type**

CanHwFilter			EcucParamConfContainerDef
BSW Description			
This container is only valid for HRHs and contains the configuration (parameters) of one hardware filter.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	**BSW Context**	
Can	Can/CanConfigSet/CanHardwareObject/CanHwFilter	
BSW Parameter		**BSW Type**
CanHwFilterCode		EcucIntegerParamDef
BSW Description		
Specifies (together with the filter mask) the identifiers range that passes the hardware filter.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
Can	Can/CanConfigSet/CanHardwareObject/CanHwFilter	
BSW Parameter		**BSW Type**
CanHwFilterMask		EcucIntegerParamDef
BSW Description		
Describes a mask for hardware-based filtering of CAN identifiers. The CAN identifiers of incoming messages are masked with the appropriate CanFilterMaskValue. Bits holding a 0 mean don't care, i.e. do not compare the message's identifier in the respective bit position. The mask shall be build by filling with leading 0. In case of CanIdType EXTENDED or MIXED a 29 bit mask shall be build. In case of CanIdType STANDARD a 11 bit mask shall be build		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
Can	Can/CanConfigSet/CanHardwareObject	
BSW Parameter		**BSW Type**
CanHwObjectCount		EcucIntegerParamDef
BSW Description		

Number of hardware objects used to implement one HOH. In case of a HRH this parameter defines the number of elements in the hardware FIFO or the number of shadow buffers, in case of a HTH it defines the number of hardware objects used for multiplexed transmission or for a hardware FIFO used by a FullCAN HTH.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
Can	Can/CanConfigSet/CanHardwareObject	
BSW Parameter		**BSW Type**
CanIdType		EcucEnumerationParamDef
BSW Description		
Specifies whether the IdValue is of type - standard identifier - extended identifier - mixed mode ImplementationType: Can_IdType		
M2 Template	**M2 Description**	
TPS_SYST	... two types of frame formats. The standard frame format uses 11-bit identifiers ... the extended frame format allows 29-bit identifiers ...	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.canAddressingMode		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Can	Can/CanConfigSet/CanHardwareObject	
BSW Parameter		**BSW Type**
CanIdValue		EcucIntegerParamDef
BSW Description		
Specifies (together with the filter mask) the identifiers range that passes the hardware filter. This parameter is set to obsolete and will be replaced by CanHwFilterCode		
M2 Template	**M2 Description**	
TPS_SYST	To describe a frames identifier on the communication system, usualy with a fixed identifierValue.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.identifier		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Can	Can/CanConfigSet/CanHardwareObject	
BSW Parameter		**BSW Type**

CanMainFunctionRWPeriodRef			EcucReferenceDef
BSW Description			
Reference to CanMainFunctionReadPeriod and CanMainFunctionWritePeriod			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**	
Can	Can/CanConfigSet/CanHardwareObject	
BSW Parameter		**BSW Type**
CanObjectId		EcucIntegerParamDef
BSW Description		
Holds the handle ID of HRH or HTH. The value of this parameter is unique in a given CAN Driver, and it should start with 0 and continue without any gaps. The HRH and HTH Ids share a common ID range. Example: HRH0-0, HRH1-1, HTH0-2, HTH1-3		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
Can	Can/CanConfigSet/CanHardwareObject	
BSW Parameter		**BSW Type**
CanObjectType		EcucEnumerationParamDef
BSW Description		
Specifies if the HardwareObject is used as Transmit or as Receive object		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
Can	Can/CanConfigSet/CanHardwareObject	
BSW Parameter		**BSW Type**
CanTTHardwareObjectTrigger		EcucParamConfContainerDef
BSW Description		

CanTTHardwareObjectTrigger is specified in the SWS TTCAN and contains the configuration (parameters) of TTCAN triggers for Hardware Objects, which are additional to the configuration (parameters) of CAN Hardware Objects. This container is only included and valid if TTCAN is supported by the controller and, enabled (see CanSupportTTCANRef, ECUC_Can_00430), and used.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
Can	Can/CanConfigSet/CanHardwareObject/CanTTHardwareObjectTrigger	
BSW Parameter		**BSW Type**
CanTTHardwareObjectBaseCycle		EcucIntegerParamDef
BSW Description		
Defines the cycle_offset. CanTTHardwareObjectBaseCycle must be not greater than cycle_count_max.		
M2 Template	**M2 Description**	
TPS_SYST	The first communication cycle where the frame is sent.This value is incremented at the beginning of each new cycle, ranging from 0 to 63, and is reset to 0 after a sequence of 64 cycles.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreTopology::CycleRepetition.BaseCycle		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Can	Can/CanConfigSet/CanHardwareObject/CanTTHardwareObjectTrigger	
BSW Parameter		**BSW Type**
CanTTHardwareObjectCycleRepetition		EcucIntegerParamDef
BSW Description		
Defines the repeat_factor. CanTTHardwareObjectCycleRepetition shall be a power of two (2), greater than cycle_offset but not greater than cycle_count_max + 1.		
M2 Template	**M2 Description**	
TPS_SYST	The number of communication cycles (after the first cycle) whenever the frame described by this timing is sent again.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreTopology::CycleRepetition.CycleRepetition		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Can	Can/CanConfigSet/CanHardwareObject/CanTTHardwareObjectTrigger	
BSW Parameter		**BSW Type**
CanTTHardwareObjectTimeMark		EcucIntegerParamDef
BSW Description		

Defines the point in time, when the trigger will be activated. Value is given in cycle time.	
M2 Template	**M2 Description**
	Time mark of trigger
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
Can	Can/CanConfigSet/CanHardwareObject/CanTTHardwareObjectTrigger
BSW Parameter	**BSW Type**
CanTTHardwareObjectTriggerId	EcucIntegerParamDef
BSW Description	
Sequential number which allows separation of different TTCAN triggers configured for one and the same hardware object.	
M2 Template	**M2 Description**
	parameter for separation of different triggers defined for one and the same hardware object
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
Can	Can/CanConfigSet/CanHardwareObject/CanTTHardwareObjectTrigger
BSW Parameter	**BSW Type**
CanTTHardwareObjectTriggerType	EcucEnumerationParamDef
BSW Description	
Defines the type of the trigger associated with the hardware object. This parameter depends on plain CAN parameter CAN_OBJECT_TYPE. If CAN_OBJECT_TYPE equals RECEIVE than this parameter is fixed to CAN_TT_RX_TRIGGER. If CAN_OBJECT_TYPE equals TRANSMIT than one of the following literals is configurable: CAN_TT_TX_REF_TRIGGER, CAN_TT_TX_REF_TRIGGER_GAP, CAN_TT_TX_TRIGGER_MERGED, CAN_TT_TX_TRIGGER_SINGLE, CAN_TT_TX_TRIGGER_EXCLUSIVE.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
Can	Can/CanConfigSet
BSW Parameter	**BSW Type**
CanIcom	EcucParamConfContainerDef
BSW Description	
This container contains the parameters for configuring pretended networking	

M2 Template	M2 Description	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanIcom	
BSW Parameter		**BSW Type**
CanIcomConfig		EcucParamConfContainerDef
BSW Description		
This container contains the configuration parameters of the ICOM Configuration.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanIcom/CanIcomConfig	
BSW Parameter		**BSW Type**
CanIcomConfigId		EcucIntegerParamDef
BSW Description		
This parameter identifies the ID of the ICOM configuration.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanIcom/CanIcomConfig	
BSW Parameter		**BSW Type**
CanIcomWakeOnBusOff		EcucBooleanParamDef
BSW Description		
This parameter defines that the MCU shall wake if the bus off is detected or not.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanIcom/CanIcomConfig	
BSW Parameter		**BSW Type**

CanIcomWakeupCauses	EcucParamConfContainerDef
BSW Description	
This container contains the configuration parameters of the wakeup causes to leave the power saving mode.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**
Can	Can/CanConfigSet/CanIcom/CanIcomConfig/CanIcomWakeupCauses
BSW Parameter	**BSW Type**
CanIcomRxMessage	EcucParamConfContainerDef
BSW Description	
This container contains the configuration parameters for the wakeup causes for matching received messages. It has to be configured as often as received messages are defined as wakeup cause. constraint: For all CanIcomRxMessage instances the Message IDs which are defined in CanIcomMessageId and in CanIcomRxMessageRange shall never overlap. Also the ranges of CanIcomRxMessageRange shall not overlap each other.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**
Can	Can/CanConfigSet/CanIcom/CanIcomConfig/CanIcomWakeupCauses/CanIcomRxMessage
BSW Parameter	**BSW Type**
CanIcomCounterValue	EcucIntegerParamDef
BSW Description	
This parameter defines that the MCU shall wake if the message with the ID is received n times on the communication channel.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**
Can	Can/CanConfigSet/CanIcom/CanIcomConfig/CanIcomWakeupCauses/CanIcomRxMessage
BSW Parameter	**BSW Type**
CanIcomMessageId	EcucIntegerParamDef
BSW Description	

This parameter defines the message ID the wakeup causes of this CanIcomRxMessage are configured for. In addition a mask can be defined,CanIcomMessageIdMask) in that case it is possible to define a range of rx messages,which can create and wakeup.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
Can	Can/CanConfigSet/CanIcom/CanIcomConfig/CanIcomWakeupCauses/CanIcomRxMessage	
BSW Parameter		**BSW Type**
CanIcomMessageIdMask		EcucIntegerParamDef
BSW Description		
Describes a mask for filtering of CAN identifiers. The CAN identifiers of incoming messages are masked with the appropriate CanIcomMessagIdMask. Bits holding a 0 mean don't care, i.e. do not compare the message's identifier in the respective bit position. The mask shall be build by filling with leading 0.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
Can	Can/CanConfigSet/CanIcom/CanIcomConfig/CanIcomWakeupCauses/CanIcomRxMessage	
BSW Parameter		**BSW Type**
CanIcomMissingMessageTimerValue		EcucFloatParamDef
BSW Description		
This parameter defines that the MCU shall wake if the message with the ID is not received for a specific time in ms on the communication channel.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
Can	Can/CanConfigSet/CanIcom/CanIcomConfig/CanIcomWakeupCauses/CanIcomRxMessage	
BSW Parameter		**BSW Type**
CanIcomPayloadLengthError		EcucBooleanParamDef
BSW Description		
This parameter defines that the MCU shall wake if a payload error occurs		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanIcom/CanIcomConfig/CanIcomWakeupCauses/CanIcomRxMessage	
BSW Parameter		**BSW Type**
CanIcomRxMessageSignalConfig		EcucParamConfContainerDef
BSW Description		
This container contains the configuration parameters for the wakeup causes for matching signals. It has to be configured as often as a signal is defined as wakeup cause. If at least one the Signal conditions evaluate to true, the whole wakeup condition is considered to be true. All instances of this container refer to the same frame/pdu		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanIcom/CanIcomConfig/CanIcomWakeupCauses/CanIcomRxMessage/CanIcomRxMessageSignalConfig	
BSW Parameter		**BSW Type**
CanIcomSignalMask		EcucParamConfContainerDef
BSW Description		
This container defines the configuration for the signal mask to extract the signal out of the payload of the received message. The mask shall be logical ANDed with the received payload. The result shall be used with the CanIcomSignalOperation with the specified parameters in CanIcomSignalValue		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanIcom/CanIcomConfig/CanIcomWakeupCauses/CanIcomRxMessage/CanIcomRxMessageSignalConfig/CanIcomSignalMask	
BSW Parameter		**BSW Type**
CanIcomSignalMaskIndex		EcucIntegerParamDef
BSW Description		
This parameter defines the byte number of the payload data to identifiy the signal. The signal can be at least one bit and in maximum 8 bits. If the signal is greater than 8 bits more than one CanIcomSignalMaskIndex must be defined. The CanIcomSignalMaskIndex works in combination with the CanIcomSignalMaskValue.		

M2 Template	M2 Description	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanIcom/CanIcomConfig/CanIcomWakeupCauses/CanIcomRxMessage/CanIcomRxMessageSignalConfig/CanIcomSignalMask	
BSW Parameter		**BSW Type**
CanIcomSignalMaskValue		EcucIntegerParamDef
BSW Description		
This parameter shall be used to mask a signal in the payload of a CAN message. The mask is logical AND with the payload. The result will be used in combination of the operations defined in CanIcomSignalOperation with the CanIcomSignalValue.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanIcom/CanIcomConfig/CanIcomWakeupCauses/CanIcomRxMessage/CanIcomRxMessageSignalConfig	
BSW Parameter		**BSW Type**
CanIcomSignalOperation		EcucEnumerationParamDef
BSW Description		
This parameter defines the operation, which shall be used to verify the payload data. AND: The masked payload via CanIcomSignalMask AND CanIcomSignalValue must be TRUE (logical AND) OR: The masked payload via CanIcomSignalMask OR CanIcomSignalValue must be TRUE (logical OR) XOR: The masked payload via CanIcomSignalMask XOR CanIcomSignalValue must be TRUE (logical XOR) SMALLER: The masked payload via CanIcomSignalMask must be strictly smaller than CanIcom-SignalValue EQUAL: The masked payload via CanIcomSignalMask must beequal to CanIcomSignalValue GREATER: The masked payload via CanIcomSignalMask must be strictly greater than CanIcom-SignalValue GREATER and SMALLER can only be applied to unsigned integer values. It is known that AND and EQUAL can be considered the same, but for clarification reason both are kept to show that AND is taken for the arithmetical and EQUAL is taken for the logical operation.		
M2 Template	**M2 Description**	
M2 Parameter		

Mapping Rule	Mapping Type

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanIcom/CanIcomConfig/CanIcomWakeupCauses/CanIcomRxMessage/CanIcomRxMessageSignalConfig	
BSW Parameter		**BSW Type**
CanIcomSignalRef		EcucReferenceDef
BSW Description		
This parameter defines a reference to the signal which shall be checked additional to the message id. This reference is used for documentation to define which ComSignal originates this filter setting. All signals being refered by this reference shall point to the same PDU.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanIcom/CanIcomConfig/CanIcomWakeupCauses/CanIcomRxMessage/CanIcomRxMessageSignalConfig	
BSW Parameter		**BSW Type**
CanIcomSignalValue		EcucParamConfContainerDef
BSW Description		
This container defines the configuration for the signal value which shall be used to compare with the signal in the received message.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanIcom/CanIcomConfig/CanIcomWakeupCauses/CanIcomRxMessage/CanIcomRxMessageSignalConfig/CanIcomSignalValue	
BSW Parameter		**BSW Type**
CanIcomSignalValueIndex		EcucIntegerParamDef
BSW Description		
This parameter defines the byte number of the value, which shall be used to compare with the received signal in combination with the CanIcomSignalOperation.The signal can be at least one bit and in maximum 8 bits. If the signal is greater than 8 bits more than one CanIcomSignalValueIndex must be defined. The CanIcomSignalValueIndex works in combination with the CanIcomSignalValueValue.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
Can	Can/CanConfigSet/CanIcom/CanIcomConfig/CanIcomWakeupCauses/CanIcomRxMessage/CanIcomRxMessageSignalConfig/CanIcomSignalValue	
BSW Parameter		**BSW Type**
CanIcomSignalValueValue		EcucIntegerParamDef
BSW Description		
This parameter shall be used to define a signal which shall be used in combination with the received signal in the payload of a CAN message. The parameter will be used in combination of the operations defined in CanIcomSignalOperation with the result of the payload and the CanIcomSignalMask.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
Can	Can	
BSW Parameter		**BSW Type**
CanGeneral		EcucParamConfContainerDef
BSW Description		
This container contains the parameters related each CAN Driver Unit.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Can	Can/CanGeneral	
BSW Parameter		**BSW Type**
CanChangeBaudrateApi		EcucBooleanParamDef
BSW Description		
The support of the Can_ChangeBaudrate API is optional. If this parameter is set to true the Can_ChangeBaudrate API shall be supported. Otherwise the API is not supported. Please note that the Can_ChangeBaudrate API and this parameter are deprecated and will be removed in future.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context		
Can	Can/CanGeneral		
BSW Parameter		**BSW Type**	
CanCounterRef		EcucReferenceDef	
BSW Description			
This parameter contains a reference to the counter, which is used by the CAN driver.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
Can	Can/CanGeneral		
BSW Parameter		**BSW Type**	
CanDevErrorDetection		EcucBooleanParamDef	
BSW Description			
Switches the Development Error Detection and Notification ON or OFF.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
Can	Can/CanGeneral		
BSW Parameter		**BSW Type**	
CanHardwareCancellation		EcucBooleanParamDef	
BSW Description			
Specifies if hardware cancellation shall be supported.ON or OFF			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
Can	Can/CanGeneral		
BSW Parameter		**BSW Type**	
CanIcomGeneral		EcucParamConfContainerDef	
BSW Description			
This container contains the general configuration parameters of the ICOM Configuration.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	BSW Context		
Can	Can/CanGeneral/CanIcomGeneral		
BSW Parameter		**BSW Type**	
CanIcomLevel		EcucEnumerationParamDef	
BSW Description			
Defines the level of Pretended Networking. This parameter is reserved for future implementations (Pretended Networking level 2).			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	BSW Context		
Can	Can/CanGeneral/CanIcomGeneral		
BSW Parameter		**BSW Type**	
CanIcomVariant		EcucEnumerationParamDef	
BSW Description			
Defines the variant, which is supported by this CanController			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	BSW Context		
Can	Can/CanGeneral		
BSW Parameter		**BSW Type**	
CanIdenticalIdCancellation		EcucBooleanParamDef	
BSW Description			
Enables/disables cancellation of pending PDUs with identical ID.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context	
Can	Can/CanGeneral	
BSW Parameter		**BSW Type**
CanIndex		EcucIntegerParamDef
BSW Description		
Specifies the InstanceId of this module instance. If only one instance is present it shall have the Id 0.		

M2 Template	M2 Description
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context
Can	Can/CanGeneral
BSW Parameter	**BSW Type**
CanLPduReceiveCalloutFunction	EcucFunctionNameDef
BSW Description	
This parameter defines the existence and the name of a callout function that is called after a successful reception of a received CAN Rx L-PDU. If this parameter is omitted no callout shall take place.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context
Can	Can/CanGeneral
BSW Parameter	**BSW Type**
CanMainFunctionBusoffPeriod	EcucFloatParamDef
BSW Description	
This parameter describes the period for cyclic call to Can_MainFunction_Busoff. Unit is seconds.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context
Can	Can/CanGeneral
BSW Parameter	**BSW Type**
CanMainFunctionModePeriod	EcucFloatParamDef
BSW Description	
This parameter describes the period for cyclic call to Can_MainFunction_Mode. Unit is seconds.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context

Can	Can/CanGeneral	
BSW Parameter		**BSW Type**
CanMainFunctionRWPeriods		EcucParamConfContainerDef
BSW Description		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
Can	Can/CanGeneral/CanMainFunctionRWPeriods	
BSW Parameter		**BSW Type**
CanMainFunctionReadPeriod		EcucFloatParamDef
BSW Description		
This parameter describes the period for cyclic call to Can_MainFunction_Read. Unit is seconds. Different poll-cycles will be configurable if more than one CanMainFunctionReadPeriod is configured. In this case multiple Can_MainFunction_Read() will be provided by the CAN Driver module.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
Can	Can/CanGeneral/CanMainFunctionRWPeriods	
BSW Parameter		**BSW Type**
CanMainFunctionWritePeriod		EcucFloatParamDef
BSW Description		
This parameter describes the period for cyclic call to Can_MainFunction_Write. Unit is seconds. Different poll-cycles will be configurable if more than one CanMainFunctionWritePeriod is configured. In this case multiple Can_MainFunction_Write() will be provided by the CAN Driver module.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
Can	Can/CanGeneral	
BSW Parameter		**BSW Type**
CanMainFunctionWakeupPeriod		EcucFloatParamDef
BSW Description		
This parameter describes the period for cyclic call to Can_MainFunction_Wakeup. Unit is seconds.		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**	
Can	Can/CanGeneral	
BSW Parameter		**BSW Type**
CanMultiplexedTransmission		EcucBooleanParamDef
BSW Description		
Specifies if multiplexed transmission shall be supported.ON or OFF		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
Can	Can/CanGeneral	
BSW Parameter		**BSW Type**
CanPublicIcomSupport		EcucBooleanParamDef
BSW Description		
Selects support of Pretended Network features in Can driver. True: Enabled False: Disabled		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
Can	Can/CanGeneral	
BSW Parameter		**BSW Type**
CanSetBaudrateApi		EcucBooleanParamDef
BSW Description		
The support of the Can_SetBaudrate API is optional. If this parameter is set to true the Can_SetBaudrate API shall be supported. Otherwise the API is not supported.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**

Can	Can/CanGeneral		
BSW Parameter		**BSW Type**	
CanSupportTTCANRef		EcucReferenceDef	
BSW Description			
The parameter refers to CanIfSupportTTCAN parameter in the CAN Interface Module configuration. The CanIfSupportTTCAN parameter defines whether TTCAN is supported.			
M2 Template	**M2 Description**		
	Defines whether TTCAN is supported or not (reference to CanIfSupportTTCAN)		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
Can	Can/CanGeneral		
BSW Parameter		**BSW Type**	
CanTimeoutDuration		EcucFloatParamDef	
BSW Description			
Specifies the maximum time for blocking function until a timeout is detected. Unit is seconds.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
Can	Can/CanGeneral		
BSW Parameter		**BSW Type**	
CanVersionInfoApi		EcucBooleanParamDef	
BSW Description			
Switches the Can_GetVersionInfo() API ON or OFF.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

D.2.2 Can Interface Mapping

BSW Module	**BSW Context**	
CanIf	CanIf	
BSW Parameter		**BSW Type**
CanIfCtrlDrvCfg		EcucParamConfContainerDef
BSW Description		
Configuration parameters for all the underlying CAN Driver modules are aggregated under this container. For each CAN Driver module a seperate instance of this container has to be provided.		

M2 Template	M2 Description	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanIf	CanIf/CanIfCtrlDrvCfg	
BSW Parameter		**BSW Type**
CanIfCtrlCfg		EcucParamConfContainerDef
BSW Description		
This container contains the configuration (parameters) of an adressed CAN controller by an underlying CAN Driver module. This container is configurable per CAN controller.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanIf	CanIf/CanIfCtrlDrvCfg/CanIfCtrlCfg	
BSW Parameter		**BSW Type**
CanIfCtrlCanCtrlRef		EcucSymbolicNameReferenceDef
BSW Description		
This parameter references to the logical handle of the underlying CAN controller from the CAN Driver module to be served by the CAN Interface module. The following parameters of CanController config container shall be referenced by this link: CanControllerId, CanWakeupSourceRef Range: 0..max. number of underlying supported CAN controllers		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanIf	CanIf/CanIfCtrlDrvCfg/CanIfCtrlCfg	
BSW Parameter		**BSW Type**
CanIfCtrlId		EcucIntegerParamDef
BSW Description		
This parameter abstracts from the CAN Driver specific parameter Controller. Each controller of all connected CAN Driver modules shall be assigned to one specific ControllerId of the CanIf. Range: 0..number of configured controllers of all CAN Driver modules		
M2 Template	**M2 Description**	
M2 Parameter		

Mapping Rule	Mapping Type
	local

BSW Module	BSW Context	
CanIf	CanIf/CanIfCtrlDrvCfg/CanIfCtrlCfg	
BSW Parameter		**BSW Type**
CanIfCtrlWakeupSupport		EcucBooleanParamDef
BSW Description		
This parameter defines if a respective controller of the referenced CAN Driver modules is queriable for wake up events. True: Enabled False: Disabled		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanIf	CanIf/CanIfCtrlDrvCfg	
BSW Parameter		**BSW Type**
CanIfCtrlDrvInitHohConfigRef		EcucReferenceDef
BSW Description		
Reference to the Init Hoh Configuration		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanIf	CanIf/CanIfCtrlDrvCfg	
BSW Parameter		**BSW Type**
CanIfCtrlDrvNameRef		EcucReferenceDef
BSW Description		
CAN Interface Driver Reference. This reference can be used to get any information (Ex. Driver Name, Vendor ID) from the CAN driver. The CAN Driver name can be derived from the ShortName of the CAN driver module.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context
CanIf	CanIf/CanIfCtrlDrvCfg
BSW Parameter	**BSW Type**
CanIfCtrlDrvTxCancellation	EcucBooleanParamDef
BSW Description	
Selects whether transmit cancellation is supported and if the appropriate callback will be provided to the CAN Driver module. True: Enabled False: Disabled	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context
CanIf	CanIf
BSW Parameter	**BSW Type**
CanIfDispatchCfg	EcucParamConfContainerDef
BSW Description	
Callback functions provided by upper layer modules of the CanIf. The callback functions defined in this container are common to all configured CAN Driver / CAN Transceiver Driver modules.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context
CanIf	CanIf/CanIfDispatchCfg
BSW Parameter	**BSW Type**
CanIfDispatchUserCheckTrcvWakeFlagIndicationName	EcucFunctionNameDef
BSW Description	
This parameter defines the name of <User_ClearTrcvWufFlagIndication>. If CANIF_DISPATCH_USERCHECKTRCVWAKEFLAGINDICATION_UL equals CAN_SM the name of <User_CheckTrcvWakeFlagIndication> is fixed. If it equals CDD, the name is selectable. If CANIF_PUBLIC_PN_SUPPORT equals False, this parameter shall not be configurable.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	BSW Context
CanIf	CanIf/CanIfDispatchCfg
BSW Parameter	**BSW Type**

CanIfDispatchUserCheckTrcvWakeFlagIndicationUL	EcucEnumerationParamDef
BSW Description	
This parameter defines the upper layer module to which the CheckTrcvWakeFlagIndication from the Driver modules have to be routed. If CANIF_PUBLIC_PN_SUPPORT equals False, this parameter shall not be configurable.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**
CanIf	CanIf/CanIfDispatchCfg
BSW Parameter	**BSW Type**
CanIfDispatchUserClearTrcvWufFlagIndicationName	EcucFunctionNameDef
BSW Description	
This parameter defines the name of <User_ClearTrcvWufFlagIndication>. If CANIF_DISPATCH_USERCLEARTRCVWUFFLAGINDICATION_UL equals CAN_SM the name of <User_ClearTrcvWufFlagIndication> is fixed. If it equals CDD, the name is selectable. If CANIF_PUBLIC_PN_SUPPORT equals False, this parameter shall not be configurable.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**
CanIf	CanIf/CanIfDispatchCfg
BSW Parameter	**BSW Type**
CanIfDispatchUserClearTrcvWufFlagIndicationUL	EcucEnumerationParamDef
BSW Description	
This parameter defines the upper layer module to which the ClearTrcvWufFlagIndication from the Driver modules have to be routed. If CANIF_PUBLIC_PN_SUPPORT equals False, this parameter shall not be configurable.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**
CanIf	CanIf/CanIfDispatchCfg
BSW Parameter	**BSW Type**
CanIfDispatchUserConfirmPnAvailabilityName	EcucFunctionNameDef
BSW Description	

This parameter defines the name of <User_ConfirmPnAvailability>. If CANIF_DISPATCH_USERCONFIRMPNAVAILABILITY_UL equals CAN_SM the name of <User_ConfirmPnAvailability> is fixed. If it equals CDD, the name is selectable. If CANIF_PUBLIC_PN_SUPPORT equals False, this parameter shall not be configurable.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**
CanIf	CanIf/CanIfDispatchCfg
BSW Parameter	**BSW Type**
CanIfDispatchUserConfirmPnAvailabilityUL	EcucEnumerationParamDef
BSW Description	
This parameter defines the upper layer module to which the ConfirmPnAvailability notification from the Driver modules have to be routed. If CANIF_PUBLIC_PN_SUPPORT equals False, this parameter shall not be configurable.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**
CanIf	CanIf/CanIfDispatchCfg
BSW Parameter	**BSW Type**
CanIfDispatchUserCtrlBusOffName	EcucFunctionNameDef
BSW Description	
This parameter defines the name of <User_ControllerBusOff>. This parameter depends on the parameter CANIF_USERCTRLBUSOFF_UL. If CANIF_USERCTRLBUSOFF_UL equals CAN_SM the name of <User_ControllerBusOff> is fixed. If CANIF_USERCTRLBUSOFF_UL equals CDD, the name of <User_ControllerBusOff> is selectable.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
CanIf	CanIf/CanIfDispatchCfg
BSW Parameter	**BSW Type**
CanIfDispatchUserCtrlBusOffUL	EcucEnumerationParamDef
BSW Description	

This parameter defines the upper layer (UL) module to which the notifications of all ControllerBusOff events from the CAN Driver modules have to be routed via <User_ControllerBusOff>. There is no possibility to configure no upper layer (UL) module as the provider of <User_ControllerBusOff>.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanIf	CanIf/CanIfDispatchCfg	
BSW Parameter		**BSW Type**
CanIfDispatchUserCtrlModeIndicationName		EcucFunctionNameDef
BSW Description		
This parameter defines the name of <User_ControllerModeIndication>. This parameter depends on the parameter CANIF_USERCTRLMODEINDICATION_UL. If CANIF_USERCTRLMODEINDICATION_UL equals CAN_SM the name of <User_ControllerModeIndication> is fixed. If CANIF_USERCTRLMODEINDICATION_UL equals CDD, the name of <User_ControllerModeIndication> is selectable.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanIf	CanIf/CanIfDispatchCfg	
BSW Parameter		**BSW Type**
CanIfDispatchUserCtrlModeIndicationUL		EcucEnumerationParamDef
BSW Description		
This parameter defines the upper layer (UL) module to which the notifications of all ControllerTransition events from the CAN Driver modules have to be routed via <User_ControllerModeIndication>.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanIf	CanIf/CanIfDispatchCfg	
BSW Parameter		**BSW Type**
CanIfDispatchUserTrcvModeIndicationName		EcucFunctionNameDef
BSW Description		

This parameter defines the name of <User_TrcvModeIndication>. This parameter depends on the parameter CANIF_USERTRCVMODEINDICATION_UL. If CANIF_USERTRCVMODEINDICATION_UL equals CAN_SM the name of <User_TrcvModeIndication> is fixed. If CANIF_USERTRCVMODEINDICATION_UL equals CDD, the name of <User_TrcvModeIndication> is selectable.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
CanIf	CanIf/CanIfDispatchCfg
BSW Parameter	**BSW Type**
CanIfDispatchUserTrcvModeIndicationUL	EcucEnumerationParamDef
BSW Description	
This parameter defines the upper layer (UL) module to which the notifications of all TransceiverTransition events from the CAN Transceiver Driver modules have to be routed via <User_TrcvModeIndication>. If no UL module is configured, no upper layer callback function will be called.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
CanIf	CanIf/CanIfDispatchCfg
BSW Parameter	**BSW Type**
CanIfDispatchUserValidateWakeupEventName	EcucFunctionNameDef
BSW Description	
This parameter defines the name of <User_ValidateWakeupEvent>. This parameter depends on the parameter CANIF_USERVALIDATEWAKEUPEVENT_UL. CANIF_USERVALIDATEWAKEUPEVENT_UL equals ECUM the name of <User_ValidateWakeupEvent> is fixed. CANIF_USERVALIDATEWAKEUPEVENT_UL equals CDD, the name of <User_ValidateWakeupEvent> is selectable. If parameter CANIF_WAKEUP_CHECK_VALIDATION_API is disabled, no <User_ValidateWakeupEvent> API can be configured.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
CanIf	CanIf/CanIfDispatchCfg
BSW Parameter	**BSW Type**
CanIfDispatchUserValidateWakeupEventUL	EcucEnumerationParamDef

BSW Description	
This parameter defines the upper layer (UL) module to which the notifications about positive former requested wake up sources have to be routed via <User_ValidateWakeupEvent>. If parameter CANIF_WAKEUP_CHECK_VALIDATION_API is disabled, this parameter cannot be configured.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context	
CanIf	CanIf	
BSW Parameter		**BSW Type**
CanIfInitCfg		EcucParamConfContainerDef
BSW Description		
This container contains the init parameters of the CAN Interface. At least one (if only on CanIf with one possible Configuration), but multiple (CanIf with different Configurations) instances of this container are possible.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg	
BSW Parameter		**BSW Type**
CanIfBufferCfg		EcucParamConfContainerDef
BSW Description		
This container contains the Txbuffer configuration. Multiple buffers with different sizes could be configured. If CanIfBufferSize (ECUC_CanIf_00834) equals 0, the CanIf Tx L-PDU only refers via this CanIfBufferCfg the corresponding CanIfHthCfg.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfBufferCfg	
BSW Parameter		**BSW Type**
CanIfBufferHthRef		EcucReferenceDef
BSW Description		

Reference to HTH, that defines the hardware object or the pool of hardware objects configured for transmission. All the CanIf Tx L-PDUs refer via the CanIfBufferCfg and this parameter to the HTHs if TxBuffering is enabled, or not. Each HTH shall not be assigned to more than one buffer.
M2 Template | **M2 Description**
M2 Parameter
Mapping Rule | **Mapping Type**
| local

BSW Module	**BSW Context**	
CanIf	CanIf/CanIfInitCfg/CanIfBufferCfg	
BSW Parameter		**BSW Type**
CanIfBufferSize		EcucIntegerParamDef
BSW Description		
This parameter defines the number of CanIf Tx L-PDUs which can be buffered in one Txbuffer. If this value equals 0, the CanIf does not perform Txbuffering for the CanIf Tx L-PDUs which are assigned to this Txbuffer. If CanIfPublicTxBuffering equals False, this parameter equals 0 for all TxBuffer. If the CanHandleType of the referred HTH equals FULL, this parameter equals 0 for this TxBuffer.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
CanIf	CanIf/CanIfInitCfg	
BSW Parameter		**BSW Type**
CanIfInitCfgSet		EcucStringParamDef
BSW Description		
Selects the CAN Interface specific configuration setup. This type of the external data structure shall contain the post build initialization data for the CAN Interface for all underlying CAN Dirvers. constant to CanIf_ConfigType		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
CanIf	CanIf/CanIfInitCfg	
BSW Parameter		**BSW Type**
CanIfInitHohCfg		EcucParamConfContainerDef
BSW Description		
This container contains the references to the configuration setup of each underlying CAN Driver.		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
CanIf	CanIf/CanIfInitCfg/CanIfInitHohCfg
BSW Parameter	**BSW Type**
CanIfHrhCfg	EcucParamConfContainerDef
BSW Description	
This container contains configuration parameters for each hardware receive object (HRH).	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
CanIf	CanIf/CanIfInitCfg/CanIfInitHohCfg/CanIfHrhCfg
BSW Parameter	**BSW Type**
CanIfHrhCanCtrlIdRef	EcucReferenceDef
BSW Description	
Reference to controller Id to which the HRH belongs to. A controller can contain one or more HRHs.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
CanIf	CanIf/CanIfInitCfg/CanIfInitHohCfg/CanIfHrhCfg
BSW Parameter	**BSW Type**
CanIfHrhCanHandleTypeRef	EcucSymbolicNameReferenceDef
BSW Description	
The parameter refers to a particular HRH object in the CAN Driver Module configuration. The type of the HRH can either be Full-CAN or Basic-CAN. The type of HRHs is defined in the CAN Driver Module and hence it is derived from CAN Driver Configuration of a Hardware Object. If BasicCAN is configured, software filtering is enabled. Please note that this reference is deprecated and is kept only for backward compatibility reasons. CanIfHrhIdSymRef shall be used instead to get the CanHandleType and CanObjectId of CAN Driver. In the next major release this reference will be deleted.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context		
CanIf	CanIf/CanIfInitCfg/CanIfInitHohCfg/CanIfHrhCfg		
BSW Parameter		**BSW Type**	
CanIfHrhIdSymRef		EcucSymbolicNameReferenceDef	
BSW Description			
The parameter refers to a particular HRH object in the CanDrv configuration (see CanHardwareObject ECUC_Can_00324). CanIf receives the following information of the CanDrv module by this reference: - CanHandleType (see ECUC_Can_00323) - CanObjectId (see ECUC_Can_00326)			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
CanIf	CanIf/CanIfInitCfg/CanIfInitHohCfg/CanIfHrhCfg		
BSW Parameter		**BSW Type**	
CanIfHrhRangeCfg		EcucParamConfContainerDef	
BSW Description			
Defines the parameters required for configurating multiple CANID ranges for a given same HRH.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
CanIf	CanIf/CanIfInitCfg/CanIfInitHohCfg/CanIfHrhCfg/CanIfHrhRangeCfg		
BSW Parameter		**BSW Type**	
CanIfHrhRangeBaseId		EcucIntegerParamDef	
BSW Description			
CAN Identifier used as base value in combination with CanIfHrhRangeMask for a masked ID range in which all CAN Ids shall pass the software filtering. The size of this parameter is limited by CanIfHrhRangeRxPduRangeCanIdType.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfInitHohCfg/CanIfHrhCfg/CanIfHrhRangeCfg	
BSW Parameter		**BSW Type**
CanIfHrhRangeMask		EcucIntegerParamDef

BSW Description	
Used as mask value in combination with CanIfHrhRangeBaseId for a masked ID range in which all CAN Ids shall pass the software filtering. The size of this parameter is limited by CanIfHrhRangeRx-PduRangeCanIdType.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
CanIf	CanIf/CanIfInitCfg/CanIfInitHohCfg/CanIfHrhCfg/CanIfHrhRangeCfg
BSW Parameter	**BSW Type**
CanIfHrhRangeRxPduLowerCanId	EcucIntegerParamDef
BSW Description	
Lower CAN Identifier of a receive CAN L-PDU for identifier range definition, in which all CAN Ids shall pass the software filtering.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
CanIf	CanIf/CanIfInitCfg/CanIfInitHohCfg/CanIfHrhCfg/CanIfHrhRangeCfg
BSW Parameter	**BSW Type**
CanIfHrhRangeRxPduRangeCanIdType	EcucEnumerationParamDef
BSW Description	
Specifies whether a configured Range of CAN Ids shall only consider standard CAN Ids or extended CAN Ids.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
CanIf	CanIf/CanIfInitCfg/CanIfInitHohCfg/CanIfHrhCfg/CanIfHrhRangeCfg
BSW Parameter	**BSW Type**
CanIfHrhRangeRxPduUpperCanId	EcucIntegerParamDef
BSW Description	
Upper CAN Identifier of a receive CAN L-PDU for identifier range definition, in which all CAN Ids shall pass the software filtering.	
M2 Template	**M2 Description**
M2 Parameter	

Mapping Rule	Mapping Type
	local

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfInitHohCfg/CanIfHrhCfg	
BSW Parameter		**BSW Type**
CanIfHrhSoftwareFilter		EcucBooleanParamDef
BSW Description		
Selects the hardware receive objects by using the HRH range/list from CAN Driver configuration to define, for which HRH a software filtering has to be performed at during receive processing. True: Software filtering is enabled False: Software filtering is enabled		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfInitHohCfg	
BSW Parameter		**BSW Type**
CanIfHthCfg		EcucParamConfContainerDef
BSW Description		
This container contains parameters related to each HTH.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfInitHohCfg/CanIfHthCfg	
BSW Parameter		**BSW Type**
CanIfHthCanCtrlIdRef		EcucReferenceDef
BSW Description		
Reference to controller Id to which the HTH belongs to. A controller can contain one or more HTHs.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfInitHohCfg/CanIfHthCfg	
BSW Parameter		**BSW Type**

CanIfHthCanHandleTypeRef		EcucSymbolicNameReferenceDef	
BSW Description			
The parameter refers to a particular HTH object in the CAN Driver Module configuration. The type of the HTH can either be Full-CAN or Basic-CAN. The type of HTHs is defined in the CAN Driver Module and hence it is derived from CAN Driver Configuration of a Hardware Object. Please note that this reference is deprecated and is kept only for backward compatibility reasons. CanIfHthIdSymRef shall be used instead to get the CanHandleType and CanObjectId of CAN Driver. In the next major release this reference will be deleted.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
CanIf	CanIf/CanIfInitCfg/CanIfInitHohCfg/CanIfHthCfg		
BSW Parameter		**BSW Type**	
CanIfHthIdSymRef		EcucSymbolicNameReferenceDef	
BSW Description			
The parameter refers to a particular HTH object in the CanDrv configuration (see CanHardwareObject ECUC_Can_00324). CanIf receives the following information of the CanDrv module by this reference: - CanHandleType (see ECUC_Can_00323) - CanObjectId (see ECUC_Can_00326)			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
CanIf	CanIf/CanIfInitCfg/CanIfInitHohCfg		
BSW Parameter		**BSW Type**	
CanIfInitRefCfgSet		EcucReferenceDef	
BSW Description			
Selects the CAN Interface specific configuration setup. This type of external data structure shall contain the post build initialization data for the CAN Interface for all underlying CAN Drivers.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**	
CanIf	CanIf/CanIfInitCfg	
BSW Parameter		**BSW Type**

CanIfMaxBufferSize		EcucIntegerParamDef	
BSW Description			
Maximum total size of all Tx buffers. This parameter is needed only in case of post-build loadable implementation using static memory allocation.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	**BSW Context**		
CanIf	CanIf/CanIfInitCfg		
BSW Parameter		**BSW Type**	
CanIfMaxRxPduCfg		EcucIntegerParamDef	
BSW Description			
Maximum number of Pdus. This parameter is needed only in case of post-build loadable implementation using static memory allocation.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	**BSW Context**		
CanIf	CanIf/CanIfInitCfg		
BSW Parameter		**BSW Type**	
CanIfMaxTxPduCfg		EcucIntegerParamDef	
BSW Description			
Maximum number of Pdus. This parameter is needed only in case of post-build loadable implementation using static memory allocation.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	**BSW Context**	
CanIf	CanIf/CanIfInitCfg	
BSW Parameter		**BSW Type**
CanIfRxPduCfg		EcucParamConfContainerDef
BSW Description		
This container contains the configuration (parameters) of each receive CAN L-PDU. The SHORT-NAME of "CanIfRxPduConfig" container itself represents the symolic name of Receive L-PDU.		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**	
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg	
BSW Parameter		**BSW Type**
CanIfRxPduCanId		EcucIntegerParamDef
BSW Description		
CAN Identifier of Receive CAN L-PDUs used by the CAN Interface. Exa: Software Filtering. This parameter is used if exactly one Can Identifier is assigned to the Pdu. If a range is assigned then the CanIfRxPduCanIdRange parameter shall be used. Range: 11 Bit For Standard CAN Identifier ... 29 Bit For Extended CAN identifier		
M2 Template	**M2 Description**	
TPS_SYST	To describe a frames identifier on the communication system, usualy with a fixed identifierValue.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.identifier		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg	
BSW Parameter		**BSW Type**
CanIfRxPduCanIdMask		EcucIntegerParamDef
BSW Description		
Identifier mask which denotes relevant bits in the CAN Identifier. This parameter defines a CAN Identifier range in an alternative way to CanIfRxPduCanIdRange. It identifies the bits of the configured CAN Identifier that must match the received CAN Identifier. Range: 11 bits for Standard CAN Identifier, 29 bits for Extended CAN Identifier.		
M2 Template	**M2 Description**	
TPS_SYST	Identifier mask which denotes the relevant bits in the CAN Identifier. Together with the identifier, this parameter defines a CAN identifier range.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.rxMask		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg	
BSW Parameter		**BSW Type**
CanIfRxPduCanIdRange		EcucParamConfContainerDef
BSW Description		
Optional container that allows to map a range of CAN Ids to one PduId.		
M2 Template	**M2 Description**	
M2 Parameter		

Mapping Rule	Mapping Type
	local

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg/CanIfRxPduCanIdRange	
BSW Parameter		**BSW Type**
CanIfRxPduCanIdRangeLowerCanId		EcucIntegerParamDef
BSW Description		
Lower CAN Identifier of a receive CAN L-PDU for identifier range definition, in which all CAN Ids are mapped to one PduId.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg/CanIfRxPduCanIdRange	
BSW Parameter		**BSW Type**
CanIfRxPduCanIdRangeUpperCanId		EcucIntegerParamDef
BSW Description		
Upper CAN Identifier of a receive CAN L-PDU for identifier range definition, in which all CAN Ids are mapped to one PduId.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg	
BSW Parameter		**BSW Type**
CanIfRxPduCanIdType		EcucEnumerationParamDef
BSW Description		
CAN Identifier of receive CAN L-PDUs used by the CAN Driver for CAN L-PDU reception.		
M2 Template	**M2 Description**	
TPS_SYST	... two types of frame formats. The standard frame format uses 11-bit identifiers ... the extended frame format allows 29-bit identifiers ...	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.canAddressingMode		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg	
BSW Parameter		**BSW Type**

CanIfRxPduDlc		EcucIntegerParamDef	
BSW Description			
Data Length code of received CAN L-PDUs used by the CAN Interface. Exa: DLC check. The data area size of a CAN L-PDU can have a range from 0 to 8 bytes.			
M2 Template	**M2 Description**		
TPS_SYST	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).		
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreCommunication::Frame.frameLength			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	**BSW Context**		
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg		
BSW Parameter		**BSW Type**	
CanIfRxPduHrhIdRef		EcucReferenceDef	
BSW Description			
The HRH to which Rx L-PDU belongs to, is referred through this parameter.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg		
BSW Parameter		**BSW Type**	
CanIfRxPduId		EcucIntegerParamDef	
BSW Description			
ECU wide unique, symbolic handle for receive CAN L-SDU. It shall fulfill ANSI/AUTOSAR definitions for constant defines. Range: 0..max. number of defined CanRxPduIds			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg		
BSW Parameter		**BSW Type**	
CanIfRxPduReadData		EcucBooleanParamDef	
BSW Description			
Enables and disables the Rx buffering for reading of received L-SDU data. True: Enabled False: Disabled			

M2 Template	M2 Description	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg	
BSW Parameter		**BSW Type**
CanIfRxPduReadNotifyStatus		EcucBooleanParamDef
BSW Description		
Enables and disables receive indication for each receive CAN L-SDU for reading its notification status. True: Enabled False: Disabled		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg	
BSW Parameter		**BSW Type**
CanIfRxPduRef		EcucReferenceDef
BSW Description		
Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg	
BSW Parameter		**BSW Type**
CanIfRxPduUserRxIndicationName		EcucFunctionNameDef
BSW Description		
This parameter defines the name of the <User_RxIndication>. This parameter depends on the parameter CANIF_RXPDU_USERRXINDICATION_UL. If CANIF_RXPDU_USERRXINDICATION_UL equals CAN_TP, CAN_NM, PDUR, XCP, J1939NM or J1939TP, the name of the <User_RxIndication> is fixed. If CANIF_RXPDU_USERRXINDICATION_UL equals CDD, the name of the <User_RxIndication> is selectable.		
M2 Template	**M2 Description**	
M2 Parameter		

Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg
BSW Parameter	**BSW Type**
CanIfRxPduUserRxIndicationUL	EcucEnumerationParamDef
BSW Description	
This parameter defines the upper layer (UL) module to which the indication of the successfully received CANRXPDUID has to be routed via <User_RxIndication>. This <User_RxIndication> has to be invoked when the indication of the configured CANRXPDUID will be received by an Rx indication event from the CAN Driver module. If no upper layer (UL) module is configured, no <User_RxIndication> has to be called in case of an Rx indication event of the CANRXPDUID from the CAN Driver module.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg
BSW Parameter	**BSW Type**
CanIfTTRxFrameTriggering	EcucParamConfContainerDef
BSW Description	
CanIfTTRxFrameTriggering is specified in the SWS TTCAN Interface and defines Frame trigger for TTCAN reception. This container is only included and valid if TTCAN is supported by the controller, enabled (see CanIfSupportTTCAN, ECUC_CanIf_00675), and a joblist is used for reception.	
M2 Template	**M2 Description**
TPS_SYST	CAN specific attributes to the FrameTriggering
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg/CanIfTTRxFrameTriggering
BSW Parameter	**BSW Type**
CanIfTTRxHwObjectTriggerIdRef	EcucReferenceDef
BSW Description	
This parameter refers to a particular TTCAN hardware receive object Trigger of a hardware object in the TTCAN Driver Module, which is referred via plain CAN parameter CANIF_HRH_HANDLETYPE_REF. This parameter is only configurable if a joblist is enabled by parameter CanIfTTJobList.	
M2 Template	**M2 Description**
	reference to a specific trigger defined for a HRH

M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**	
CanIf	CanIf/CanIfInitCfg/CanIfRxPduCfg/CanIfTTRxFrameTriggering	
BSW Parameter		**BSW Type**
CanTTRxJoblistTimeMark		EcucIntegerParamDef
BSW Description		
Defines the point in time, when the joblist execution funciton (JLEF) shall be called for the referenced rx trigger. Value is given in cycle time. This parameter is only configurable if a joblist is enabled by parameter CanIfTTJobList.		
M2 Template	**M2 Description**	
	Time mark for calling the joblist (for message processing)	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
CanIf	CanIf/CanIfInitCfg	
BSW Parameter		**BSW Type**
CanIfTxPduCfg		EcucParamConfContainerDef
BSW Description		
This container contains the configuration (parameters) of a transmit CAN L-PDU. It has to be configured as often as a transmit CAN L-PDU is needed. The SHORT-NAME of "CanIfTxPduConfig" container represents the symolic name of Transmit L-PDU.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg	
BSW Parameter		**BSW Type**
CanIfTTTxFrameTriggering		EcucParamConfContainerDef
BSW Description		
CanIfTTTxFrameTriggering is specified in the SWS TTCAN Interface and defines Frame trigger for TTCAN transmission. This container is only included and valid if TTCAN is supported by the controller, enabled (see CanIfSupportTTCAN, ECUC_CanIf_00675), and a joblist is used.		
M2 Template	**M2 Description**	
TPS_SYST	CAN specific attributes to the FrameTriggering	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering		

Mapping Rule	Mapping Type
1:1 mapping	full

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg/CanIfTTTxFrameTriggering	
BSW Parameter		**BSW Type**
CanIfTTTxHwObjectTriggerIdRef		EcucReferenceDef
BSW Description		
This parameter refers to a particular TTCAN hardware transmit object Trigger of a hardware object in the TTCAN Driver Module, which is referred via plain CAN parameter CANIF_HTH_HANDLETYPE_REF. This parameter is only configurable if a joblist is enabled by parameter CanIfTTJobList.		
M2 Template	**M2 Description**	
	reference to a specific trigger defined for a HTH	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg/CanIfTTTxFrameTriggering	
BSW Parameter		**BSW Type**
CanIfTTTxJoblistTimeMark		EcucIntegerParamDef
BSW Description		
Defines the point in time, when the joblist execution funciton (JLEF) shall be called for the referenced tx frame trigger. Value is given in cycle time. This parameter is only configurable if a joblist is enabled by parameter CanIfTTJobList.		
M2 Template	**M2 Description**	
	Time mark for calling the joblist (for message processing)	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg	
BSW Parameter		**BSW Type**
CanIfTxPduBufferRef		EcucReferenceDef
BSW Description		
Configurable reference to a CanIf buffer configuration.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context

CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg		
BSW Parameter		**BSW Type**	
CanIfTxPduCanId		EcucIntegerParamDef	
BSW Description			
CAN Identifier of transmit CAN L-PDUs used by the CAN Driver for CAN L-PDU transmission. Range: 11 Bit For Standard CAN Identifier ... 29 Bit For Extended CAN identifier The CAN Identifier may be omitted for dynamic transmit L-PDUs.			
M2 Template	**M2 Description**		
TPS_SYST	To describe a frames identifier on the communication system, usualy with a fixed identifierValue.		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.identifier			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	**BSW Context**		
CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg		
BSW Parameter		**BSW Type**	
CanIfTxPduCanIdMask		EcucIntegerParamDef	
BSW Description			
Identifier mask which denotes relevant bits in the CAN Identifier. This parameter may be used to keep parts of the CAN Identifier of dynamic transmit L-PDUs static. Range: 11 bits for Standard CAN Identifier, 29 bits for Extended CAN Identifier.			
M2 Template	**M2 Description**		
TPS_SYST	Identifier mask which denotes static bits in the CAN identifier. The other bits can be set dynamically.		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.txMask			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	**BSW Context**		
CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg		
BSW Parameter		**BSW Type**	
CanIfTxPduCanIdType		EcucEnumerationParamDef	
BSW Description			
Type of CAN Identifier of the transmit CAN L-PDU used by the CAN Driver module for CAN L-PDU transmission.			
M2 Template	**M2 Description**		
TPS_SYST	... two types of frame formats. The standard frame format uses 11-bit identifiers ... the extended frame format allows 29-bit identifiers ...		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.canAddressingMode			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	**BSW Context**		
CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg		
BSW Parameter		**BSW Type**	

CanIfTxPduDlc			EcucIntegerParamDef
BSW Description			
This parameter is set to obsolete and will be removed. It was replaced by a combination of the size of the global PDU (referenced via CanIfTxPduRef) and the parameter CanIfFixedBuffer. Old description: Data length code (in bytes) of transmit CAN L-PDUs used by the CAN Driver for CAN L-PDU transmission. The data area size of a CAN L-Pdu can have a range from 0 to 8 bytes.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	**BSW Context**		
CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg		
BSW Parameter		**BSW Type**	
CanIfTxPduId		EcucIntegerParamDef	
BSW Description			
ECU wide unique, symbolic handle for transmit CAN L-SDU. Range: 0..max. number of CantTxPduIds			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg		
BSW Parameter		**BSW Type**	
CanIfTxPduPnFilterPdu		EcucBooleanParamDef	
BSW Description			
If CanIfPublicPnFilterSupport is enabled, by this parameter PDUs could be configured which will pass the CanIfPnFilter. If there is no CanIfTxPduPnFilterPdu configured per controller, the corresponding controller applies no CanIfPnFilter.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**	
CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg	
BSW Parameter		**BSW Type**
CanIfTxPduReadNotifyStatus		EcucBooleanParamDef
BSW Description		

Enables and disables transmit confirmation for each transmit CAN L-SDU for reading its notification status. True: Enabled False: Disabled		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg	
BSW Parameter		**BSW Type**
CanIfTxPduRef		EcucReferenceDef
BSW Description		
Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg	
BSW Parameter		**BSW Type**
CanIfTxPduType		EcucEnumerationParamDef
BSW Description		
Defines the type of each transmit CAN L-PDU.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg	
BSW Parameter		**BSW Type**
CanIfTxPduUserTxConfirmationName		EcucFunctionNameDef
BSW Description		
This parameter defines the name of the <User_TxConfirmation>. This parameter depends on the parameter CANIF_TXPDU_USERTXCONFIRMATION_UL. If CANIF_TXPDU_USERTXCONFIRMATION_UL equals CAN_TP, CAN_NM, PDUR, XCP, J1939NM or J1939TP, the name of the <User_TxConfirmation> is fixed. If CANIF_TXPDU_USERTXCONFIRMATION_UL equals CDD, the name of the <User_TxConfirmation> is selectable.		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**	
CanIf	CanIf/CanIfInitCfg/CanIfTxPduCfg	
BSW Parameter		**BSW Type**
CanIfTxPduUserTxConfirmationUL		EcucEnumerationParamDef
BSW Description		
This parameter defines the upper layer (UL) module to which the confirmation of the successfully transmitted CANTXPDUID has to be routed via the <User_TxConfirmation>. This <User_TxConfirmation> has to be invoked when the confirmation of the configured CANTXPDUID will be received by a Tx confirmation event from the CAN Driver module. If no upper layer (UL) module is configured, no <User_TxConfirmation> has to be called in case of a Tx confirmation event of the CANTXPDUID from the CAN Driver module.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
CanIf	CanIf	
BSW Parameter		**BSW Type**
CanIfPrivateCfg		EcucParamConfContainerDef
BSW Description		
This container contains the private configuration (parameters) of the CAN Interface.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
CanIf	CanIf/CanIfPrivateCfg	
BSW Parameter		**BSW Type**
CanIfFixedBuffer		EcucBooleanParamDef
BSW Description		
This parameter defines if the buffer element length shall be fixed to 8 Bytes. TRUE: Buffer element length is fixed to 8 Bytes. FALSE: Buffer element length depends on the size of the referencing PDUs.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**
CanIf	CanIf/CanIfPrivateCfg
BSW Parameter	**BSW Type**
CanIfPrivateDlcCheck	EcucBooleanParamDef
BSW Description	
Selects whether the DLC check is supported. True: Enabled False: Disabled	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
CanIf	CanIf/CanIfPrivateCfg
BSW Parameter	**BSW Type**
CanIfPrivateSoftwareFilterType	EcucEnumerationParamDef
BSW Description	
Selects the desired software filter mechanism for reception only. Each implemented software filtering method is identified by this enumeration number. Range: Types implemented software filtering methods	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
CanIf	CanIf/CanIfPrivateCfg
BSW Parameter	**BSW Type**
CanIfSupportTTCAN	EcucBooleanParamDef
BSW Description	
Defines whether TTCAN is supported. TRUE: TTCAN is supported. FALSE: TTCAN is not supported, only normal CAN communication is possible.	
M2 Template	**M2 Description**
	Defines whether TTCAN is supported or not
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**
CanIf	CanIf/CanIfPrivateCfg
BSW Parameter	**BSW Type**

CanIfTTGeneral			EcucParamConfContainerDef
BSW Description			
CanIfTTGeneral is specified in the SWS TTCAN Interface and defines if and in which way TTCAN is supported. This container is only included and valid if TTCAN is supported by the controller, enabled (see CanIfSupportTTCAN, ECUC_CanIf_00675), and used.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
CanIf	CanIf/CanIfPrivateCfg/CanIfTTGeneral		
BSW Parameter		**BSW Type**	
CanIfTTDemEventParameterRefs		EcucParamConfContainerDef	
BSW Description			
Container for the references to DemEventParameter elements which shall be invoked using the API Dem_ReportErrorStatus API in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId value. The standardized errors are provided in the container and can be extended by vendor specific error references.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	**BSW Context**		
CanIf	CanIf/CanIfPrivateCfg/CanIfTTGeneral/CanIfTTDemEventParameterRefs		
BSW Parameter		**BSW Type**	
CANIF_TT_E_JLE_SYNC		EcucSymbolicNameReferenceDef	
BSW Description			
Reference to configured DEM event to report that the JLEF lost synchronization to the local time of the TTCAN controller.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	**BSW Context**	
CanIf	CanIf/CanIfPrivateCfg/CanIfTTGeneral	
BSW Parameter		**BSW Type**
CanIfTTJoblist		EcucBooleanParamDef
BSW Description		

Defines whether TTCAN is processed via a joblist. TRUE: Joblist is used. FALSE: No joblist is used. This parameter is only configurable if TTCAN is enabled by parameter CanIfSupportTTCAN.		
M2 Template	**M2 Description**	
	Defines whehter a joblist is used or not	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
CanIf	CanIf/CanIfPrivateCfg/CanIfTTGeneral	
BSW Parameter		**BSW Type**
CanIfTTMaxIsrDelay		EcucIntegerParamDef
BSW Description		
Defines the maximum delay for the execution of the joblist execution function JLEF. This parameter is only configurable if a joblist is enabled by parameter CanIfTTJobList.		
M2 Template	**M2 Description**	
	Defines the maximum delay for the job list execution	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
CanIf	CanIf	
BSW Parameter		**BSW Type**
CanIfPublicCfg		EcucParamConfContainerDef
BSW Description		
This container contains the public configuration (parameters) of the CAN Interface.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
CanIf	CanIf/CanIfPublicCfg	
BSW Parameter		**BSW Type**
CanIfMetaDataSupport		EcucBooleanParamDef
BSW Description		
Enable support for dynamic ID handling using L-SDU MetaData.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

	local

BSW Module	BSW Context	
CanIf	CanIf/CanIfPublicCfg	
BSW Parameter		**BSW Type**
CanIfPublicCancelTransmitSupport		EcucBooleanParamDef
BSW Description		
Configuration parameter to enable/disable dummy API for upper layer modules which allows to request the cancellation of an I-PDU.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanIf	CanIf/CanIfPublicCfg	
BSW Parameter		**BSW Type**
CanIfPublicCddHeaderFile		EcucStringParamDef
BSW Description		
Defines header files for callback functions which shall be included in case of CDDs. Range of characters is 1.. 32.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanIf	CanIf/CanIfPublicCfg	
BSW Parameter		**BSW Type**
CanIfPublicChangeBaudrateSupport		EcucBooleanParamDef
BSW Description		
Configuration parameter to enable/disable the API to change the baudrate of a CAN controller. True: Enabled False: Disabled Please note that the CanIf_ChangeBaudrate API and this parameter are deprecated and will be removed in future.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context

CanIf	CanIf/CanIfPublicCfg
BSW Parameter	**BSW Type**
CanIfPublicDevErrorDetect	EcucBooleanParamDef
BSW Description	
Enables and disables the development error detection and notification mechanism. True: Enabled False: Disabled	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
CanIf	CanIf/CanIfPublicCfg
BSW Parameter	**BSW Type**
CanIfPublicHandleTypeEnum	EcucEnumerationParamDef
BSW Description	
This parameter is used to configure the Can_HwHandleType. The Can_HwHandleType represents the hardware object handles of a CAN hardware unit. For CAN hardware units with more than 255 HW objects the extended range shall be used (UINT16).	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
CanIf	CanIf/CanIfPublicCfg
BSW Parameter	**BSW Type**
CanIfPublicIcomSupport	EcucBooleanParamDef
BSW Description	
Selects support of Pretended Network features in CanIf. True: Enabled False: Disabled	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**
CanIf	CanIf/CanIfPublicCfg
BSW Parameter	**BSW Type**
CanIfPublicMultipleDrvSupport	EcucBooleanParamDef
BSW Description	

Selects support for multiple CAN Drivers. True: Enabled False: Disabled		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
CanIf	CanIf/CanIfPublicCfg	
BSW Parameter		**BSW Type**
CanIfPublicNumberOfCanHwUnits		EcucIntegerParamDef
BSW Description		
This parameter is set to obsolete and will be removed. It is not required because the same information is available by checking the actual multiplicity of CanIfCtrlDrvCfg. Old description: Number of served CAN hardware units.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
CanIf	CanIf/CanIfPublicCfg	
BSW Parameter		**BSW Type**
CanIfPublicPnSupport		EcucBooleanParamDef
BSW Description		
Selects support of Partial Network features in CanIf. True: Enabled False: Disabled		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
CanIf	CanIf/CanIfPublicCfg	
BSW Parameter		**BSW Type**
CanIfPublicReadRxPduDataApi		EcucBooleanParamDef
BSW Description		
Enables / Disables the API CanIf_ReadRxPduData() for reading received L-SDU data. True: Enabled False: Disabled		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**	
CanIf	CanIf/CanIfPublicCfg	
BSW Parameter		**BSW Type**
CanIfPublicReadRxPduNotifyStatusApi		EcucBooleanParamDef
BSW Description		
Enables and disables the API for reading the notification status of receive L-PDUs. True: Enabled False: Disabled		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
CanIf	CanIf/CanIfPublicCfg	
BSW Parameter		**BSW Type**
CanIfPublicReadTxPduNotifyStatusApi		EcucBooleanParamDef
BSW Description		
Enables and disables the API for reading the notification status of transmit L-PDUs. True: Enabled False: Disabled		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
CanIf	CanIf/CanIfPublicCfg	
BSW Parameter		**BSW Type**
CanIfPublicSetDynamicTxIdApi		EcucBooleanParamDef
BSW Description		
Enables and disables the API for reconfiguration of the CAN Identifier for each Transmit L-PDU. True: Enabled False: Disabled		
M2 Template	**M2 Description**	
M2 Parameter		

Mapping Rule	Mapping Type
	local

BSW Module	BSW Context	
CanIf	CanIf/CanIfPublicCfg	
BSW Parameter		**BSW Type**
CanIfPublicTxBuffering		EcucBooleanParamDef
BSW Description		
Enables and disables the buffering of transmit L-PDUs (rejected by the CanDrv) within the CAN Interface module. True: Enabled False: Disabled		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanIf	CanIf/CanIfPublicCfg	
BSW Parameter		**BSW Type**
CanIfPublicTxConfirmPollingSupport		EcucBooleanParamDef
BSW Description		
Configuration parameter to enable/disable the API to poll for Tx Confirmation state.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanIf	CanIf/CanIfPublicCfg	
BSW Parameter		**BSW Type**
CanIfPublicVersionInfoApi		EcucBooleanParamDef
BSW Description		
Enables and disables the API for reading the version information about the CAN Interface. True: Enabled False: Disabled		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanIf	CanIf/CanIfPublicCfg	
BSW Parameter		**BSW Type**
CanIfPublicWakeupCheckValidByNM		EcucBooleanParamDef
BSW Description		
If enabled, only NM messages shall validate a detected wake-up event (see CANIF722) at the corresponding wake-up source in the CanIf. If disabled, all messages shall validate such a wake-up event. This parameter depends on CANIF_PUBLIC_WAKEUP_CHECK_VALID_API and shall only be configurable, if it is enabled. True: Enabled False: Disabled		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanIf	CanIf/CanIfPublicCfg	
BSW Parameter		**BSW Type**
CanIfPublicWakeupCheckValidSupport		EcucBooleanParamDef
BSW Description		
Selects support for wake up validation True: Enabled False: Disabled		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanIf	CanIf/CanIfPublicCfg	
BSW Parameter		**BSW Type**
CanIfSetBaudrateApi		EcucBooleanParamDef
BSW Description		
Configuration parameter to enable/disable the CanIf_SetBaudrate API to change the baud rate of a CAN Controller. If this parameter is set to true the CanIf_SetBaudrate API shall be supported. Otherwise the API is not supported.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanIf	CanIf/CanIfPublicCfg	
BSW Parameter		**BSW Type**
CanIfTxOfflineActiveSupport		EcucBooleanParamDef
BSW Description		
Determines wether TxOffLineActive feature (see SWS_CANIF_00072) is supported by CanIf. True: Enabled False: Disabled		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanIf	CanIf	
BSW Parameter		**BSW Type**
CanIfTrcvDrvCfg		EcucParamConfContainerDef
BSW Description		
This container contains the configuration (parameters) of all addressed CAN transceivers by each underlying CAN Transceiver Driver module. For each CAN transceiver Driver a seperate instance of this container shall be provided.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanIf	CanIf/CanIfTrcvDrvCfg	
BSW Parameter		**BSW Type**
CanIfTrcvCfg		EcucParamConfContainerDef
BSW Description		
This container contains the configuration (parameters) of one addressed CAN transceiver by the underlying CAN Transceiver Driver module. For each CAN transceiver a seperate instance of this container has to be provided.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanIf	CanIf/CanIfTrcvDrvCfg/CanIfTrcvCfg	
BSW Parameter		**BSW Type**
CanIfTrcvCanTrcvRef		EcucSymbolicNameReferenceDef
BSW Description		

This parameter references to the logical handle of the underlying CAN transceiver from the CAN transceiver driver module to be served by the CAN Interface module. Range: 0..max. number of underlying supported CAN transceivers		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
CanIf	CanIf/CanIfTrcvDrvCfg/CanIfTrcvCfg	
BSW Parameter		**BSW Type**
CanIfTrcvId		EcucIntegerParamDef
BSW Description		
This parameter abstracts from the CAN Transceiver Driver specific parameter Transceiver. Each transceiver of all connected CAN Transceiver Driver modules shall be assigned to one specific TransceiverId of the CanIf. Range: 0..number of configured transceivers of all CAN Transceiver Driver modules		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
CanIf	CanIf/CanIfTrcvDrvCfg/CanIfTrcvCfg	
BSW Parameter		**BSW Type**
CanIfTrcvWakeupSupport		EcucBooleanParamDef
BSW Description		
This parameter defines if a respective transceiver of the referenced CAN Transceiver Driver modules is queriable for wake up events. True: Enabled False: Disabled		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

D.2.3 Can Transceiver Mapping

BSW Module	**BSW Context**	
CanTrcv	CanTrcv	
BSW Parameter		**BSW Type**

CanTrcvConfigSet		EcucParamConfContainerDef	
BSW Description			
This is the multiple configuration set container for CAN Transceiver.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	**BSW Context**		
CanTrcv	CanTrcv/CanTrcvConfigSet		
BSW Parameter		**BSW Type**	
CanTrcvChannel		EcucParamConfContainerDef	
BSW Description			
Container gives CAN transceiver driver information about a single CAN transceiver (channel).			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	**BSW Context**		
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel		
BSW Parameter		**BSW Type**	
CanTrcvAccess		EcucChoiceContainerDef	
BSW Description			
Container gives CanTrcv Driver information about access to a single CAN transceiver.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	**BSW Context**		
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess		
BSW Parameter		**BSW Type**	
CanTrcvDioAccess		EcucParamConfContainerDef	
BSW Description			
Container gives CAN transceiver driver information about accessing ports and port pins. In addition relation between CAN transceiver hardware pin names and Dio port access information is given. If a CAN transceiver hardware has no Dio interface, there is no instance of this container.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	BSW Context
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvDioAccess
BSW Parameter	**BSW Type**
CanTrcvDioChannelAccess	EcucParamConfContainerDef
BSW Description	
Container gives DIO channel access by single Can transceiver channel.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	BSW Context
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvDioAccess/CanTrcvDioChannelAccess
BSW Parameter	**BSW Type**
CanTrcvDioSymNameRef	EcucChoiceReferenceDef
BSW Description	
Choice Reference to a DIO Port, DIO Channel or DIO Channel Group. This reference replaces the CANTRCV_DIO_PORT_SYM_NAME, CANTRCV_DIO_CHANNEL_SYM_NAME and CANTRCV_DIO_GROUP_SYM_NAME references in the Can Trcv SWS.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	BSW Context
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvDioAccess/CanTrcvDioChannelAccess
BSW Parameter	**BSW Type**
CanTrcvHardwareInterfaceName	EcucStringParamDef
BSW Description	
CAN transceiver hardware interface name. It is typically the name of a pin. From a Dio point of view it is either a port, a single channel or a channel group. Depending on this fact either CANTRCV_DIO_PORT_SYMBOLIC_NAME or CANTRCV_DIO_CHANNEL_SYMBOLIC_NAME or CANTRCV_DIO_CHANNEL_GROUP_SYMBOLIC_NAME shall reference a Dio configuration. The CAN transceiver driver implementation description shall list up this name for the appropriate CAN transceiver hardware.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess	
BSW Parameter		**BSW Type**
CanTrcvSpiAccess		EcucParamConfContainerDef
BSW Description		
Container gives CAN transceiver driver information about accessing Spi. If a CAN transceiver hardware has no Spi interface, there is no instance of this container.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvSpiAccess	
BSW Parameter		**BSW Type**
CanTrcvSpiSequence		EcucParamConfContainerDef
BSW Description		
Container gives CAN transceiver driver information about one SPI sequence. One SPI sequence used by CAN transceiver driver is in exclusive use for it. No other driver is allowed to access this sequence. CAN transceiver driver may use one sequence to access n CAN transceiver hardwares chips of the same type or n sequences are used to access one single CAN transceiver hardware chip. If a CAN transceiver hardware has no SPI interface, there is no instance of this container.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvSpiAccess/CanTrcvSpiSequence	
BSW Parameter		**BSW Type**
CanTrcvSpiAccessSynchronous		EcucBooleanParamDef
BSW Description		
This parameter is used to define whether the access to the Spi sequence is synchronous or asynchronous. true: SPI access is synchronous. false: SPI access is asynchronous.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvAccess/CanTrcvSpiAccess/CanTrcvSpiSequence	
BSW Parameter		**BSW Type**
CanTrcvSpiSequenceName		EcucSymbolicNameReferenceDef
BSW Description		
Reference to a Spi sequence configuration container.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel	
BSW Parameter		**BSW Type**
CanTrcvChannelId		EcucIntegerParamDef
BSW Description		
Unique identifier of the CAN Transceiver Channel.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel	
BSW Parameter		**BSW Type**
CanTrcvChannelUsed		EcucBooleanParamDef
BSW Description		
Shall the related CAN transceiver channel be used?		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel	
BSW Parameter		**BSW Type**
CanTrcvControlsPowerSupply		EcucBooleanParamDef
BSW Description		
Is ECU power supply controlled by this transceiver? TRUE = Controlled by transceiver. FALSE = Not controlled by transceiver.		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel	
BSW Parameter		**BSW Type**
CanTrcvHwPnSupport		EcucBooleanParamDef
BSW Description		
Indicates whether the HW supports the selective wake-up function TRUE = Selective wakeup feature is supported by the transceiver FALSE = Selective wakeup functionality is not available in transceiver		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel	
BSW Parameter		**BSW Type**
CanTrcvInitState		EcucEnumerationParamDef
BSW Description		
State of CAN transceiver after call to CanTrcv_Init.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel	
BSW Parameter		**BSW Type**
CanTrcvMaxBaudrate		EcucIntegerParamDef
BSW Description		
Max baudrate for transceiver hardware type. Only used for validation purposes. Value shall be configured by configuration tool based on transceiver hardware type.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel	
BSW Parameter		**BSW Type**
CanTrcvPartialNetwork		EcucParamConfContainerDef
BSW Description		
Container gives CAN transceiver driver information about the configuration of Partial Networking functionality.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork	
BSW Parameter		**BSW Type**
CanTrcvBaudRate		EcucIntegerParamDef
BSW Description		
Indicates the CAN Bus communication baud rate in kbps.		
M2 Template	**M2 Description**	
TPS_SYST		
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreTopology::CommunicationCluster.speed		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork	
BSW Parameter		**BSW Type**
CanTrcvBusErrFlag		EcucBooleanParamDef
BSW Description		
Indicates if the Bus Error (BUSERR) flag is managed by the BSW. This flag is set if a bus failure is detected by the transceiver. TRUE = Supported by transceiver and managed by BSW. FALSE = Not managed by BSW.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork	
BSW Parameter		**BSW Type**
CanTrcvPnCanIdIsExtended		EcucBooleanParamDef
BSW Description		
Indicates whether extended or standard ID is used. TRUE = Extended Can identifier is used. FALSE = Standard Can identifier is used		

M2 Template	**M2 Description**	
TPS_SYST		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanCommunicationConnector.pncWakeupCanIdExtended		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork	
BSW Parameter		**BSW Type**
CanTrcvPnEnabled		EcucBooleanParamDef
BSW Description		
Indicates whether the selective wake-up function is enabled or disabled in HW. TRUE = Selective wakeup feature is enabled in the transceiver hardware FALSE = Selective wakeup feature is disabled in the transceiver hardware		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork	
BSW Parameter		**BSW Type**
CanTrcvPnFrameCanId		EcucIntegerParamDef
BSW Description		
CAN ID of the Wake-up Frame (WUF).		
M2 Template	**M2 Description**	
TPS_SYST		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanCommunicationConnector.pncWakeupCanId		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork	
BSW Parameter		**BSW Type**
CanTrcvPnFrameCanIdMask		EcucIntegerParamDef
BSW Description		
ID Mask for the selective activation of the transceiver. It is used to enableFrame Wake-up (WUF) on a group of IDs.		
M2 Template	**M2 Description**	
TPS_SYST		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanCommunicationConnector.pncWakeupCanIdMask		

Mapping Rule	Mapping Type
1:1 mapping	full

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork	
BSW Parameter		**BSW Type**
CanTrcvPnFrameDataMaskSpec		EcucParamConfContainerDef
BSW Description		
Defines data payload mask to be used on the received payload in order to determine if the transceiver must be woken up by the received Wake-up Frame (WUF).		
M2 Template	**M2 Description**	
TPS_SYST		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanCommunicationConnector.pncWakeupDataMask		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/CanTrcvPnFrameDataMaskSpec	
BSW Parameter		**BSW Type**
CanTrcvPnFrameDataMask		EcucIntegerParamDef
BSW Description		
Defines the n byte (Byte0 = LSB) of the data payload mask to be used on the received payload in order to determine if the transceiver must be woken up by the received Wake-up Frame (WUF).		
M2 Template	**M2 Description**	
TPS_SYST		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanCommunicationConnector.pncWakeupDataMask		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork/CanTrcvPnFrameDataMaskSpec	
BSW Parameter		**BSW Type**
CanTrcvPnFrameDataMaskIndex		EcucIntegerParamDef
BSW Description		
holds the position n in frame of the mask-part		
M2 Template	**M2 Description**	
TPS_SYST		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanCommunicationConnector.pncWakeupDataMask		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context

CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork
BSW Parameter	**BSW Type**
CanTrcvPnFrameDlc	EcucIntegerParamDef
BSW Description	
Data Length of the Wake-up Frame (WUF).	
M2 Template	**M2 Description**
TPS_SYST	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanCommunicationConnector.pncWakeupDlc	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel/CanTrcvPartialNetwork
BSW Parameter	**BSW Type**
CanTrcvPowerOnFlag	EcucBooleanParamDef
BSW Description	
Description: Indicates if the Power On Reset (POR) flag is available and is managed by the transceiver. TRUE = Supported by Hardware. FALSE = Not supported by Hardware	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel
BSW Parameter	**BSW Type**
CanTrcvPorWakeupSourceRef	EcucSymbolicNameReferenceDef
BSW Description	
Symbolic name reference to specify the wakeup sources that should be used in the calls to EcuM_SetWakeupEvent as specified in [SWS_CanTrcv_00183] and [SWS_CanTrcv_00184]. This reference is mandatory if the HW supports POR or SYSERR flags	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel
BSW Parameter	**BSW Type**
CanTrcvSyserrWakeupSourceRef	EcucSymbolicNameReferenceDef
BSW Description	

Symbolic name reference to specify the wakeup sources that should be used in the calls to EcuM_SetWakeupEvent as specified in [SWS_CanTrcv_00183] and [SWS_CanTrcv_00184] This reference is mandatory if the HW supports POR or SYSERR flags	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel	
BSW Parameter		**BSW Type**
CanTrcvWakeupByBusUsed		EcucBooleanParamDef
BSW Description		
Is wake up by bus supported? If CAN transceiver hardware does not support wake up by bus value is always FALSE. If CAN transceiver hardware supports wake up by bus value is TRUE or FALSE depending whether it is used or not. TRUE = Is used. FALSE = Is not used.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
CanTrcv	CanTrcv/CanTrcvConfigSet/CanTrcvChannel	
BSW Parameter		**BSW Type**
CanTrcvWakeupSourceRef		EcucReferenceDef
BSW Description		
Reference to a wakeup source in the EcuM configuration. This reference is only needed if CanTrcvWakeupByBusUsed is true.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
CanTrcv	CanTrcv/CanTrcvConfigSet	
BSW Parameter		**BSW Type**
CanTrcvSPICommRetries		EcucIntegerParamDef
BSW Description		
Indicates the maximum number of communication retries in case of a failed SPI communication (applies both to timed out communication and to errors/NACK in the response data). If configured value is ’0’, no retry is allowed (communication is expected to succeed at first try).		

M2 Template	M2 Description	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvConfigSet	
BSW Parameter		**BSW Type**
CanTrcvSPICommTimeout		EcucIntegerParamDef
BSW Description		
Indicates the maximum time allowed to the CanTrcv for replying (either positively or negatively) to a SPI command. Timeout is configured in milliseconds. Timeout value of '0' means that no specific timeout is to be used by CanTrcv and the communication is executed at the best of the SPI HW capacity.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
CanTrcv	CanTrcv	
BSW Parameter		**BSW Type**
CanTrcvGeneral		EcucParamConfContainerDef
BSW Description		
Container gives CAN transceiver driver basic information.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvGeneral	
BSW Parameter		**BSW Type**
CanTrcvDevErrorDetect		EcucBooleanParamDef
BSW Description		
Switches development error detection and notification on and off. If switched on, #define CANTRCV_DEV _ERROR_DETECT ON shall be generated. If switched off, #define CANTRCV_DEV_ERROR _DETECT OFF shall be generated. Define shall be part of file CanTrcv_Cfg.h.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvGeneral	
BSW Parameter		**BSW Type**
CanTrcvGetVersionInfo		EcucBooleanParamDef
BSW Description		
Switches version information API on and off. If switched off, function need not be present in compiled code.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvGeneral	
BSW Parameter		**BSW Type**
CanTrcvSPICommRetries		EcucIntegerParamDef
BSW Description		
Indicates the maximal number of communication retries in case of failed SPI communication (applies both to timed out communication and to errors/NACK in the response data). (0 ... 255 times, 0 means no retry allowed, communication must succeed at first try)		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvGeneral	
BSW Parameter		**BSW Type**
CanTrcvSPICommTimeout		EcucIntegerParamDef
BSW Description		
Indicates the maximal time allowed to the Transceiver in order to reply (either positively or negatively) to a SPI command. (value in ms, 0ms means no specific timeout is to be used, communication is executed at the best of the SPI HW capacity)		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
CanTrcv	CanTrcv/CanTrcvGeneral	
BSW Parameter		**BSW Type**

CanTrcvWaitCount		EcucIntegerParamDef	
BSW Description			
Indicates the number of wait states to change the transceiver operation mode. Transceiver hardware may need wait states for some transitions.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	**BSW Context**		
CanTrcv	CanTrcv/CanTrcvGeneral		
BSW Parameter		**BSW Type**	
CanTrcvWakeUpSupport		EcucEnumerationParamDef	
BSW Description			
Informs whether wake up is supported by polling or not supported. In case no wake up is supported by the hardware, setting has to be NOT_SUPPORTED. Only in the case of wake up supported by polling, function CanTrcv_MainFunction has to be present and to be invoked by the scheduler.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

D.2.4 CanNm Mapping

BSW Module	**BSW Context**		
CanNm	CanNm		
BSW Parameter		**BSW Type**	
CanNmGlobalConfig		EcucParamConfContainerDef	
BSW Description			
This container contains the global configuration parameter of the CanNm. The parameters and the parameters of the sub containers shall be mapped to the C data type CanNm_ConfigType (for parameters where it is possible) which is passed to the CanNm_Init function. This container is a MultipleConfigurationContainer (only for variant 3), i.e. this container and its sub-containers exit once per configuration set.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**	
CanNm	CanNm/CanNmGlobalConfig	
BSW Parameter		**BSW Type**

CanNmBusLoadReductionEnabled		EcucBooleanParamDef	
BSW Description			
Pre-processor switch for enabling busload reduction support.			
M2 Template	**M2 Description**		
TPS_SYST	Enables busload reduction support		
M2 Parameter			
SystemTemplate::NetworkManagement::CanNmClusterCoupling.nmBusloadReductionEnabled			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	**BSW Context**		
CanNm	CanNm/CanNmGlobalConfig		
BSW Parameter		**BSW Type**	
CanNmBusSynchronizationEnabled		EcucBooleanParamDef	
BSW Description			
Pre-processor switch for enabling bus synchronization support. This feature is required for gateway nodes only.			
M2 Template	**M2 Description**		
TPS_SYST	Enables bus synchronization support.		
M2 Parameter			
SystemTemplate::NetworkManagement::NmEcu.nmBusSynchronizationEnabled			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	**BSW Context**		
CanNm	CanNm/CanNmGlobalConfig		
BSW Parameter		**BSW Type**	
CanNmChannelConfig		EcucParamConfContainerDef	
BSW Description			
This container contains the channel specific configuration parameter of the CanNm.			
M2 Template	**M2 Description**		
TPS_SYST	Can specific NmCluster attributes		
M2 Parameter			
SystemTemplate::NetworkManagement::CanNmCluster			
Mapping Rule			**Mapping Type**
Create container for each existing CanNmCluster.			full

BSW Module	**BSW Context**		
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig		
BSW Parameter		**BSW Type**	
CanNmActiveWakeupBitEnabled		EcucBooleanParamDef	
BSW Description			
Enables/Disables the handling of the Active Wakeup Bit in the CanNm module.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig	
BSW Parameter		**BSW Type**
CanNmAllNmMessagesKeepAwake		EcucBooleanParamDef
BSW Description		
Specifies if CanNm drops irrelevant NM PDUs. false: Only NM PDUs with a PNI bit = true and containing a PN request for this ECU triggers the standard RX indication handling true: Every NM PDU triggers the standard RX indication handling		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig	
BSW Parameter		**BSW Type**
CanNmBusLoadReductionActive		EcucBooleanParamDef
BSW Description		
This parameter defines if bus load reduction for the respective NM channel is active or not.		
M2 Template	**M2 Description**	
TPS_SYST	It determines if bus load reduction for the respective CanNm channel is active or not.	
M2 Parameter		
SystemTemplate::NetworkManagement::CanNmCluster.nmBusloadReductionActive		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig	
BSW Parameter		**BSW Type**
CanNmCarWakeUpBitPosition		EcucIntegerParamDef
BSW Description		
Specifies the Bit position of the CWU within the NM PDU.		
M2 Template	**M2 Description**	
TPS_SYST	Specifies the bit position of the CarWakeUp within the NM PDU.	
M2 Parameter		
SystemTemplate::NetworkManagement::CanNmCluster.nmCarWakeUpBitPosition		
Mapping Rule		**Mapping Type**
The position of the Car Wakeup bit in the Ecuc is defined by the configuration parameters CanNmCarWakeUpBytePosition and CanNmCarWakeUpBitPosition (position in wakeUpByte). In the SysT the position is described only by the bit position in the NmMessage.		full

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig	
BSW Parameter		**BSW Type**

CanNmCarWakeUpBytePosition		EcucIntegerParamDef	
BSW Description			
Specifies the Byte position of the CWU within the NM PDU.			
M2 Template	**M2 Description**		
TPS_SYST	Specifies the bit position of the CarWakeUp within the NM PDU.		
M2 Parameter			
SystemTemplate::NetworkManagement::CanNmCluster.nmCarWakeUpBitPosition			
Mapping Rule			**Mapping Type**
The position of the Car Wakeup bit in the Ecuc is defined by the configuration parameters CanNmCarWakeUpBytePosition and CanNmCarWakeUpBitPosition (position in wakeUpByte). In the SysT the position is described only by the bit position in the NM PDU.			full

BSW Module	**BSW Context**		
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig		
BSW Parameter		**BSW Type**	
CanNmCarWakeUpFilterEnabled		EcucBooleanParamDef	
BSW Description			
If CWU filtering is supported, only the CWU bit within the NM PDU with source node identifier CanNmCarWakeUpFilterNodeId is considered as CWU request. FALSE - CWU filtering is not supported TRUE - CWU filtering is supported			
M2 Template	**M2 Description**		
TPS_SYST	If this attribute is set to true the CareWakeUp filtering is supported. In this case only the CarWakeUp bit within the NM PDU with source node identifier nmCarWakeUpFilterNodeId is considered as CarWakeUp request.		
M2 Parameter			
SystemTemplate::NetworkManagement::CanNmCluster.nmCarWakeUpFilterEnabled			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	**BSW Context**		
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig		
BSW Parameter		**BSW Type**	
CanNmCarWakeUpFilterNodeId		EcucIntegerParamDef	
BSW Description			
Source node identifier for CWU filtering. If CWU filtering is supported, only the CWU bit within the NM PDU with source node identifier CanNmCarWakeUpFilterNodeId is considered as CWU request.			
M2 Template	**M2 Description**		
TPS_SYST	Source node identifier for CarWakeUp filtering. If CarWakeUp filtering is supported (nmCarWakeUpFilterEnabled), only the CarWakeUp bit within the NM PDU with source node identifier nmCarWakeUpFilterNodeId is considered as CarWakeUp request.		
M2 Parameter			
SystemTemplate::NetworkManagement::CanNmCluster.nmCarWakeUpFilterNodeId			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	**BSW Context**	
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig	
BSW Parameter		**BSW Type**

CanNmCarWakeUpRxEnabled		EcucBooleanParamDef	
BSW Description			
Enables or disables support of CarWakeUp bit evaluation in received NM PDUs. FALSE - CarWakeUp not supported TRUE - CarWakeUp supported			
M2 Template	**M2 Description**		
TPS_SYST	If set to true this attribute enables the support of CarWakeUp bit evaluation in received NM PDUs.		
M2 Parameter			
SystemTemplate::NetworkManagement::CanNmCluster.nmCarWakeUpRxEnabled			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	**BSW Context**		
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig		
BSW Parameter		**BSW Type**	
CanNmComMNetworkHandleRef		EcucSymbolicNameReferenceDef	
BSW Description			
This reference points to the unique channel defined by the ComMChannel and provides access to the unique channel index value in ComMChannelId.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig		
BSW Parameter		**BSW Type**	
CanNmImmediateNmCycleTime		EcucFloatParamDef	
BSW Description			
Defines the immediate NM PDU cycle time in seconds which is used for CanNmImmediateNmTransmissions NM PDU transmissions.			
M2 Template	**M2 Description**		
TPS_SYST	Timeout of a NM PDU in seconds. It determines how long the NM shall wait with notification of transmission failure while communication errors occur on the bus.		
M2 Parameter			
SystemTemplate::NetworkManagement::CanNmCluster.nmImmediateNmCycleTime			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	**BSW Context**	
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig	
BSW Parameter		**BSW Type**
CanNmImmediateNmTransmissions		EcucIntegerParamDef
BSW Description		
Defines the number of immediate NM PDUs which shall be transmitted. If the value is zero no immediate NM PDUs are transmitted. The cycle time of immeditate NM PDUs is defined by CanNmImmediateNmCycleTime.		

M2 Template	M2 Description
TPS_SYST	Defines the number of immediate NM PDUs which shall be transmitted. If the value is zero no immediate NM PDUs are transmitted. The cycle time of immediate NM PDUs is defined by nmImmediateNmCycleTime.
M2 Parameter	
SystemTemplate::NetworkManagement::CanNmCluster.nmImmediateNmTransmissions	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	**BSW Type**
CanNmMsgCycleOffset	EcucFloatParamDef
BSW Description	
Time offset in the periodic transmission node. It determines the start delay of the transmission. Specified in seconds.	
M2 Template	**M2 Description**
TPS_SYST	Node specific time offset in the periodic transmission node. It determines the start delay of the transmission. Specified in seconds.
M2 Parameter	
SystemTemplate::NetworkManagement::CanNmNode.nmMsgCycleOffset	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	**BSW Type**
CanNmMsgCycleTime	EcucFloatParamDef
BSW Description	
Period of a NM PDU 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".	
M2 Template	**M2 Description**
TPS_SYST	Period of a CanNm PDU 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.
M2 Parameter	
SystemTemplate::NetworkManagement::CanNmCluster.nmMsgCycleTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	**BSW Type**
CanNmMsgReducedTime	EcucFloatParamDef
BSW Description	
Node specific bus cycle time in the periodic transmission mode with bus load reduction. Specified in seconds.	
M2 Template	**M2 Description**
TPS_SYST	Node specific bus cycle time in the periodic transmission mode with bus load reduction. Specified in seconds.

M2 Parameter	
SystemTemplate::NetworkManagement::CanNmNode.nmMsgReducedTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**	
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig	
BSW Parameter		**BSW Type**
CanNmMsgTimeoutTime		EcucFloatParamDef
BSW Description		
Transmission Timeout of NM PDU. If there is no transmission confirmation by the CAN Interface within this timeout, the CANNM module shall give an error notification.		
M2 Template	**M2 Description**	
TPS_SYST	Timeout of a NM PDU in seconds. It determines how long the NM shall wait with notification of transmission failure while communication errors occur on the bus.	
M2 Parameter		
SystemTemplate::NetworkManagement::CanNmCluster.nmMessageTimeoutTime		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig	
BSW Parameter		**BSW Type**
CanNmNodeId		EcucIntegerParamDef
BSW Description		
Node identifier of local node.		
M2 Template	**M2 Description**	
TPS_SYST	Node identifier of local NmNode. Must be unique in the NmCluster.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmNode.nmNodeId		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig	
BSW Parameter		**BSW Type**
CanNmPduCbvPosition		EcucEnumerationParamDef
BSW Description		
Defines the position of the control bit vector within the NM PDU. The value of the parameter represents the location of the control bit vector in the NM PDU (CanNmPduByte0 means byte 0, CanNmPduByte1 means byte 1, CanNmPduOff means source node identifier is not part of the NM PDU) ImplementationType: CanNm_PduPositionType		
M2 Template	**M2 Description**	
TPS_SYST	Defines the position of the control bit vector within the NM PDU (Bitpositon).	
M2 Parameter		
SystemTemplate::NetworkManagement::CanNmCluster.nmCbvPosition		
Mapping Rule		**Mapping Type**

Derive byte position from nmCbvPosition attribute. If this optional attribute is missing set CANNM_PDU_OFF as value.	full

BSW Module	**BSW Context**		
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig		
BSW Parameter		**BSW Type**	
CanNmPduNidPosition		EcucEnumerationParamDef	
BSW Description			
Defines the position of the source node identifier within the NM PDU. The value of the parameter represents the location of the source node identifier in the NM PDU (CanNMPduByte0 means byte 0, CanNmPduByte1 means byte 1, CanNmPduOff means source node identifier is not part of the NM PDU) ImplementationType: CanNm_PduPositionType			
M2 Template	**M2 Description**		
TPS_SYST	Defines the bitposition of the source node identifier within the NM PDU.		
M2 Parameter			
SystemTemplate::NetworkManagement::CanNmCluster.nmNidPosition			
Mapping Rule			**Mapping Type**
Derive byte position from nmNidPosition attribute. If this optional attribute is missing set CANNM_PDU_OFF as value.			full

BSW Module	**BSW Context**		
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig		
BSW Parameter		**BSW Type**	
CanNmPnEnabled		EcucBooleanParamDef	
BSW Description			
Enables or disables support of partial networking. false: Partial networking Range not supported true: Partial networking supported			
M2 Template	**M2 Description**		
TPS_SYST	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.		
M2 Parameter			
SystemTemplate::PncMapping::PncMapping			
Mapping Rule			**Mapping Type**
For every EcuInstance that references an ISignalIPduGroup that is referenced by a PncMapping the CanNmPnEnabled shall be set to true.			full

BSW Module	**BSW Context**	
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig	
BSW Parameter		**BSW Type**
CanNmPnEraCalcEnabled		EcucBooleanParamDef
BSW Description		
Specifies if CanNm calculates the PN request information for external requests. (ERA) false: PN request are not calculated true: PN request are calculated		

M2 Template	M2 Description	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig	
BSW Parameter		**BSW Type**
CanNmPnEraRxNSduRef		EcucReferenceDef
BSW Description		
Reference to a Pdu in the COM-Stack. The SduRef is required for every CanNm Channel, because ERA is reported per channel.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig	
BSW Parameter		**BSW Type**
CanNmPnHandleMultipleNetworkRequests		EcucBooleanParamDef
BSW Description		
Specifies if CanNm performs an additional transition from Network Mode to Repeat Message State (true) or not (false).		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig	
BSW Parameter		**BSW Type**
CanNmRemoteSleepIndTime		EcucFloatParamDef
BSW Description		
Timeout for Remote Sleep Indication. It defines the time in seconds how long it shall take to recognize that all other nodes are ready to sleep.		
M2 Template	**M2 Description**	
TPS_SYST	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.	
M2 Parameter		
SystemTemplate::NetworkManagement::CanNmCluster.nmRemoteSleepIndicationTime		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	**BSW Type**
CanNmRepeatMessageTime	EcucFloatParamDef
BSW Description	
Timeout for Repeat Message State. It defines the time in seconds how long the NM shall stay in the Repeat Message State.	
M2 Template	**M2 Description**
TPS_SYST	Timeout for Repeat Message State in seconds. Defines the time how long the NM shall stay in the Repeat Message State.
M2 Parameter	
SystemTemplate::NetworkManagement::CanNmCluster.nmRepeatMessageTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	**BSW Type**
CanNmRxPdu	EcucParamConfContainerDef
BSW Description	
This container is used to configure the Rx PDU properties that are used for the CanNm Channel.	
M2 Template	**M2 Description**
TPS_SYST	Receive NM Pdu
M2 Parameter	
SystemTemplate::NetworkManagement::NmNode.rxNmPdu	
Mapping Rule	**Mapping Type**
Create container for each NmPdu that is received on the regarded Nm cluster	full

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig/CanNmRxPdu
BSW Parameter	**BSW Type**
CanNmRxPduId	EcucIntegerParamDef
BSW Description	
This parameter defines the Rx PDU ID of the CanIf L-PDU range that is associated with this CanNm channel.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig/CanNmRxPdu
BSW Parameter	**BSW Type**
CanNmRxPduRef	EcucReferenceDef
BSW Description	
Reference to the global PDU that is used by this CanNm channel.	
M2 Template	**M2 Description**

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**	
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig	
BSW Parameter		**BSW Type**
CanNmTimeoutTime		EcucFloatParamDef
BSW Description		
Network Timeout for NM PDUs. It denotes the time in seconds how long the NM shall stay in the Ready Sleep State before transition into the Prepare Bus-Sleep Mode is initiated.		
M2 Template	**M2 Description**	
TPS_SYST	Network Timeout for CanNm PDUs 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.	
M2 Parameter		
SystemTemplate::NetworkManagement::CanNmCluster.nmNetworkTimeout		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig	
BSW Parameter		**BSW Type**
CanNmTxPdu		EcucParamConfContainerDef
BSW Description		
This container contains the CanNmTxConfirmationPduId and the CanNmTxPduRef.		
M2 Template	**M2 Description**	
TPS_SYST	Transmit NM Pdu	
M2 Parameter		
SystemTemplate::NetworkManagement::NmNode.txNmPdu		
Mapping Rule		**Mapping Type**
Create container for each NmPdu that is transmitted on the regarded Nmcluster		full

BSW Module	**BSW Context**	
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig/CanNmTxPdu	
BSW Parameter		**BSW Type**
CanNmTxConfirmationPduId		EcucIntegerParamDef
BSW Description		
Handle Id to be used by the Lower Layer to confirm the transmission of the CanNmTxPdu to the LowerLayer.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig/CanNmTxPdu
BSW Parameter	**BSW Type**
CanNmTxPduRef	EcucReferenceDef
BSW Description	
The reference to the common PDU structure.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	**BSW Type**
CanNmUserDataLength	EcucIntegerParamDef
BSW Description	
Defines the length of the user data contained in the NM PDU Please note that this parameter is deprecated and will be removed in future.	
M2 Template	**M2 Description**
TPS_SYST	Defines the length of the user data contained in the NM Pdu.
M2 Parameter	
SystemTemplate::NetworkManagement::CanNmCluster.nmUserDataLength	
Mapping Rule	**Mapping Type**
1:1 mapping (please note that this upstream mapping is deprecated)	full

BSW Module	**BSW Context**
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	**BSW Type**
CanNmUserDataTxPdu	EcucParamConfContainerDef
BSW Description	
This optional container is used to configure the UserNm PDU. This container is only available if CanNmComUserDataSupport is enabled.	
M2 Template	**M2 Description**
TPS_SYST	This optional aggregation is used to describe NmUserData that is transmitted in the NmPdu.
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::ISignalToIPduMapping	
Mapping Rule	**Mapping Type**
Create container for each NmPdu that aggregates the ISignalToIPduMapping element. The configuration for these Pdus (e.g. Transfer Properties) shall be derived from this information.	full

BSW Module	**BSW Context**
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig/CanNmUserDataTxPdu
BSW Parameter	**BSW Type**
CanNmTxUserDataPduId	EcucIntegerParamDef
BSW Description	

This parameter defines the Handle ID of the NM User Data I-PDU.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig/CanNmUserDataTxPdu
BSW Parameter	**BSW Type**
CanNmTxUserDataPduRef	EcucReferenceDef
BSW Description	
Reference to the NM User Data I-PDU in the global PDU collection.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
CanNm	CanNm/CanNmGlobalConfig/CanNmChannelConfig
BSW Parameter	**BSW Type**
CanNmWaitBusSleepTime	EcucFloatParamDef
BSW Description	
Timeout for bus calm down phase. It denotes the time in seconds how long the NM shall stay in the Prepare Bus-Sleep Mode before transition into Bus-Sleep Mode shall take place.	
M2 Template	**M2 Description**
TPS_SYST	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.
M2 Parameter	
SystemTemplate::NetworkManagement::CanNmCluster.nmWaitBusSleepTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
CanNm	CanNm/CanNmGlobalConfig
BSW Parameter	**BSW Type**
CanNmComControlEnabled	EcucBooleanParamDef
BSW Description	
Pre-processor switch for enabling the Communication Control support.	
M2 Template	**M2 Description**
TPS_SYST	Enables the Communication Control support.
M2 Parameter	
SystemTemplate::NetworkManagement::NmEcu.nmComControlEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	BSW Context		
CanNm	CanNm/CanNmGlobalConfig		
BSW Parameter		**BSW Type**	
CanNmComUserDataSupport		EcucBooleanParamDef	
BSW Description			
Enable/disable the user data support.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
CanNm	CanNm/CanNmGlobalConfig		
BSW Parameter		**BSW Type**	
CanNmCoordinatorSyncSupport		EcucBooleanParamDef	
BSW Description			
Enables/disables the coordinator synchronisation support.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
CanNm	CanNm/CanNmGlobalConfig		
BSW Parameter		**BSW Type**	
CanNmDevErrorDetect		EcucBooleanParamDef	
BSW Description			
Pre-processor switch for enabling development error detection support.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig	
BSW Parameter		**BSW Type**
CanNmImmediateRestartEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling the asynchronous transmission of a NM PDU upon bus-communication request in Prepare-Bus-Sleep mode.		
M2 Template	**M2 Description**	
TPS_SYST	Enables the asynchronous transmission of a CanNm PDU upon bus-communication request in Prepare-Bus-Sleep mode.	
M2 Parameter		

SystemTemplate::NetworkManagement::CanNmClusterCoupling.nmImmediateRestartEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
CanNm	CanNm/CanNmGlobalConfig
BSW Parameter	**BSW Type**
CanNmImmediateTxconfEnabled	EcucBooleanParamDef
BSW Description	
Enable/disable the immediate tx confirmation.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
CanNm	CanNm/CanNmGlobalConfig
BSW Parameter	**BSW Type**
CanNmMainFunctionPeriod	EcucFloatParamDef
BSW Description	
Call cycle in seconds of CanNm_MainFunction.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
CanNm	CanNm/CanNmGlobalConfig
BSW Parameter	**BSW Type**
CanNmNodeDetectionEnabled	EcucBooleanParamDef
BSW Description	
Precompile time switch to enable the node detection feature.	
M2 Template	**M2 Description**
TPS_SYST	Enables the Request Repeat Message Request support. Only valid if nmNodeIdEnabled is set to true.
M2 Parameter	
SystemTemplate::NetworkManagement::NmEcu.nmNodeDetectionEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
CanNm	CanNm/CanNmGlobalConfig
BSW Parameter	**BSW Type**
CanNmNodeIdEnabled	EcucBooleanParamDef
BSW Description	

Pre-processor switch for enabling the source node identifier.		
M2 Template	**M2 Description**	
TPS_SYST		
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmNodeIdEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
CanNm	CanNm/CanNmGlobalConfig	
BSW Parameter		**BSW Type**
CanNmNumberOfChannels		EcucIntegerParamDef
BSW Description		
Number of Can NM channels allowed within one ECU. Please note that this parameter is deprecated and will be removed in future.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
CanNm	CanNm/CanNmGlobalConfig	
BSW Parameter		**BSW Type**
CanNmPassiveModeEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling support of the Passive Mode.		
M2 Template	**M2 Description**	
TPS_SYST	Enables support of the Passive Mode.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmPassiveModeEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
CanNm	CanNm/CanNmGlobalConfig	
BSW Parameter		**BSW Type**
CanNmPduRxIndicationEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling the PDU Rx Indication.		
M2 Template	**M2 Description**	
TPS_SYST	Switch for enabling the PDU Rx Indication.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmPduRxIndicationEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig	
BSW Parameter		**BSW Type**
CanNmPnEiraCalcEnabled		EcucBooleanParamDef
BSW Description		
Specifies if CanNm calculates the PN request information for internal an external requests. (EIRA) true: PN request are calculated false: PN request are not calculated		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig	
BSW Parameter		**BSW Type**
CanNmPnEiraRxNSduRef		EcucReferenceDef
BSW Description		
Reference to a Pdu in the COM-Stack. Only one SduRef is required for CanNm because the EIRA is the aggregation over all Can Channels.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig	
BSW Parameter		**BSW Type**
CanNmPnInfo		EcucParamConfContainerDef
BSW Description		
PN information configuration		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig/CanNmPnInfo	
BSW Parameter		**BSW Type**
CanNmPnFilterMaskByte		EcucParamConfContainerDef
BSW Description		
PN information configuration		
M2 Template	**M2 Description**	
TPS_SYST		

M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanCommunicationConnector.pncWakeupDataMask	
Mapping Rule	**Mapping Type**
For one EcuInstance all contributing pncWakeupDataMask will be bitwise ORed to obtain aggregated pncWakeupDataMask value for this ECU. Since the pncWakeupDataMask is calculated over the whole payload (8 Byte) of the Nm Pdu, the leading Bytes of this aggregated pncWakeupDataMask shall be ignored based on the System.pncVectorOffset value. In order to get the CanNmPnFilterMaskByteIndex and CanNmPnFilterMaskByteValue for all the bytes aggregated pncWakeupDataMask shall be processed in a littleEndian way. E.g. if pncVectorOffset = 2 and aggregated pncWakeupDataMask has the value 2ˆ 63 this will end up in a CanNmPnFilterMaskByte with CanNmPnFilterMaskByteIndex = 5 and CanNmPnFilterMaskByteValue = 128.	full

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig/CanNmPnInfo/CanNmPnFilterMaskByte	
BSW Parameter		**BSW Type**
CanNmPnFilterMaskByteIndex		EcucIntegerParamDef
BSW Description		
Index of the filter mask byte. Specifies the position within the filter mask byte array.		
M2 Template	**M2 Description**	
TPS_SYST		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanCommunicationConnector.pncWakeupDataMask		
Mapping Rule		**Mapping Type**
For one EcuInstance all contributing pncWakeupDataMask will be bitwise ORed to obtain aggregated pncWakeupDataMask value for this ECU. Since the pncWakeupDataMask is calculated over the whole payload (8 Byte) of the Nm Pdu, the leading Bytes of this aggregated pncWakeupDataMask shall be ignored based on the System.pncVectorOffset value. In order to get the CanNmPnFilterMaskByteIndex and CanNmPnFilterMaskByteValue for all the bytes aggregated pncWakeupDataMask shall be processed in a littleEndian way. E.g. if pncVectorOffset = 2 and aggregated pncWakeupDataMask has the value 2ˆ 63 this will end up in a CanNmPnFilterMaskByte with CanNmPnFilterMaskByteIndex = 5 and CanNmPnFilterMaskByteValue = 128.		full

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig/CanNmPnInfo/CanNmPnFilterMaskByte	
BSW Parameter		**BSW Type**
CanNmPnFilterMaskByteValue		EcucIntegerParamDef
BSW Description		
Parameter to configure the filter mask byte.		
M2 Template	**M2 Description**	
TPS_SYST		
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanTopology::CanCommunicationConnector.pncWakeupDataMask		
Mapping Rule		**Mapping Type**

For one EcuInstance all contributing pncWakeupDataMask will be bitwise ORed to obtain aggregated pncWakeupDataMask value for this ECU. Since the pncWakeupDataMask is calculated over the whole payload (8 Byte) of the NmPdu, the leading Bytes of this aggregated pncWakeupDataMask shall be ignored based on the System.pncVectorOffset value. In order to get the CanNmPnFilterMaskByteIndex and CanNmPnFilterMaskByteValue for all the bytes aggregated pncWakeupDataMask shall be processed in a littleEndian way. E.g. if pncVectorOffset = 2 and aggregated pncWakeupDataMask has the value 2ˆ 63 this will end up in a CanNmPnFilterMaskByte with CanNmPnFilterMaskByteIndex = 5 and CanNmPnFilterMaskByteValue = 128.	full

BSW Module	**BSW Context**
CanNm	CanNm/CanNmGlobalConfig/CanNmPnInfo
BSW Parameter	**BSW Type**
CanNmPnInfoLength	EcucIntegerParamDef
BSW Description	
Specifies the length of the PN request information in the NM PDU.	
M2 Template	**M2 Description**
TPS_SYST	Length of the partial networking request release information vector.
M2 Parameter	
SystemTemplate::System.pncVectorLength	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
CanNm	CanNm/CanNmGlobalConfig/CanNmPnInfo
BSW Parameter	**BSW Type**
CanNmPnInfoOffset	EcucIntegerParamDef
BSW Description	
Specifies the offset of the PN request information in the NM PDU.	
M2 Template	**M2 Description**
TPS_SYST	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.
M2 Parameter	
SystemTemplate::System.pncVectorOffset	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
CanNm	CanNm/CanNmGlobalConfig
BSW Parameter	**BSW Type**
CanNmPnResetTime	EcucFloatParamDef
BSW Description	
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. The value shall be the same for every channel. Thus it is a global config parameter.	
M2 Template	**M2 Description**
M2 Parameter	

Mapping Rule	Mapping Type
	local

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig	
BSW Parameter		**BSW Type**
CanNmRemoteSleepIndEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling remote sleep indication support. This feature is required for gateway nodes only.		
M2 Template	**M2 Description**	
TPS_SYST	Switch for enabling remote sleep indication support.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmRemoteSleepIndEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig	
BSW Parameter		**BSW Type**
CanNmRepeatMsgIndEnabled		EcucBooleanParamDef
BSW Description		
Enable/disable the notification that a RepeatMessageRequest bit has been received.		
M2 Template	**M2 Description**	
TPS_SYST	Enable/disable the notification that a RepeatMessageRequest bit has been received.	
M2 Parameter		
SystemTemplate::NetworkManagement::CanNmEcu.nmRepeatMsgIndicationEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig	
BSW Parameter		**BSW Type**
CanNmStateChangeIndEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling the CAN NM state change notification.		
M2 Template	**M2 Description**	
TPS_SYST	Enables the CAN Network Management state change notification.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmStateChangeIndEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
CanNm	CanNm/CanNmGlobalConfig	
BSW Parameter		**BSW Type**
CanNmUserDataEnabled		EcucBooleanParamDef
BSW Description		

Pre-processor switch for enabling user data support.		
M2 Template	**M2 Description**	
TPS_SYST	Switch for enabling user data support.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmUserDataEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
CanNm	CanNm/CanNmGlobalConfig	
BSW Parameter		**BSW Type**
CanNmVersionInfoApi		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling version info API support.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

D.2.5 CanTp Mapping

BSW Module	**BSW Context**	
CanTp	CanTp	
BSW Parameter		**BSW Type**
CanTpConfig		EcucParamConfContainerDef
BSW Description		
This container contains the configuration parameters and sub containers of the AUTOSAR CanTp module. This container is a MultipleConfigurationContainer, i.e. this container and its sub-containers exist once per configuration set.		
M2 Template	**M2 Description**	
TPS_SYST	This element defines exactly one CAN TP Configuration.	
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpConfig		
Mapping Rule		**Mapping Type**
Create Container if CanTpConfig exists in ECU Extract.		full

BSW Module	**BSW Context**	
CanTp	CanTp/CanTpConfig	
BSW Parameter		**BSW Type**
CanTpChannel		EcucParamConfContainerDef
BSW Description		
This container contains the configuration parameters of the CanTp channel.		
M2 Template	**M2 Description**	
TPS_SYST	Configuration parameters of the CanTp channel.	
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpChannel		
Mapping Rule		**Mapping Type**
Create Container ifor each CanTpChannel that exist in ECU Extract.		full

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel	
BSW Parameter		**BSW Type**
CanTpChannelMode		EcucEnumerationParamDef
BSW Description		
The CAN Transport Layer supports half and full duplex channel modes.		
M2 Template	**M2 Description**	
TPS_SYST	The CAN Transport Layer supports half and full duplex channel modes.	
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpChannel.channelMode		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel	
BSW Parameter		**BSW Type**
CanTpRxNSdu		EcucParamConfContainerDef
BSW Description		
The following parameters needs to be configured for each CAN N-SDU that the CanTp module receives via the CanTpChannel.		
M2 Template	**M2 Description**	
TPS_SYST	Reference to the IPdu that is segmented by the Transport Protocol.	
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpConnection.tpSdu		
Mapping Rule		**Mapping Type**
Create container for each existing CanTpConnection that contains a reference to an N-SDU that is received.		full

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu	
BSW Parameter		**BSW Type**
CanTpBs		EcucIntegerParamDef
BSW Description		
Sets the 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.		
M2 Template	**M2 Description**	
TPS_SYST	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.	
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpConnection.maxBlockSize		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu	
BSW Parameter		**BSW Type**
CanTpNAe		EcucParamConfContainerDef
BSW Description		

This container is required for each RxNSdu and TxNSdu with AddressingFormat CANTP_MIXED or CANTP_MIXED29BIT.		
M2 Template	**M2 Description**	
TPS_SYST	Declares which communication addressing mode is supported.	
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpConnection.addressingFormat		
Mapping Rule		**Mapping Type**
Create container if addressingFormat is set to "mixed".		full

BSW Module	**BSW Context**	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu/CanTpNAe	
BSW Parameter		**BSW Type**
CanTpNAe		EcucIntegerParamDef
BSW Description		
This parameter contains the transport protocol address extension value.		
M2 Template	**M2 Description**	
TPS_SYST	An ECUs TP address on the referenced channel. This represents the diagnostic Address.	
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpAddress.tpAddressExtensionValue		
Mapping Rule		**Mapping Type**
The CanTPConnection contains a reference to the SDU and a relation to the Tp Node that contains the TpAddressExtension.		full

BSW Module	**BSW Context**	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu	
BSW Parameter		**BSW Type**
CanTpNSa		EcucParamConfContainerDef
BSW Description		
This container is required for each RxNSdu and TxNSdu with RxTaType CANTP_PHYSICAL and CanTpAddressingFormat CANTP_EXTENDED. When DynIdSupport is enabled, this container is also required for each TxNSdu with AddressingFormat CANTP_NORMALFIXED or CANTP_MIXED29BIT. When DynIdSupport is enabled and GenericConnectionSupport is not enabled, this container is also required for each RxNSdu with AddressingFormat CANTP_NORMALFIXED or CANTP_MIXED29BIT.		
M2 Template	**M2 Description**	
TPS_SYST	Declares which communication addressing mode is supported.	
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpConnection.addressingFormat		
Mapping Rule		**Mapping Type**
Create container if addressingFormat is set to "extended".		full

BSW Module	**BSW Context**	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu/CanTpNSa	
BSW Parameter		**BSW Type**
CanTpNSa		EcucIntegerParamDef
BSW Description		
This parameter contains the transport protocol source address value.		
M2 Template	**M2 Description**	
TPS_SYST	An ECUs TP address on the referenced channel. This represents the diagnostic Address.	

M2 Parameter	
SystemTemplate::TransportProtocols::CanTpAddress.tpAddress	
Mapping Rule	**Mapping Type**
The CanTPConnection contains a reference to the SDU and a relation to the Tp Node that contains the TpAddress.	full

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu	
BSW Parameter		**BSW Type**
CanTpNTa		EcucParamConfContainerDef
BSW Description		
This container is required for each RxNSdu and TxNSdu with AddressingFormat CANTP_EXTENDED. When DynIdSupport is enabled, this container is also required for each RxNSdu with AddressingFormat CANTP_NORMALFIXED or CANTP_MIXED29BIT. When DynIdSupport is enabled and GenericConnectionSupport is not enabled, this container is also required for each TxNSdu with AddressingFormat CANTP_NORMALFIXED or CANTP_MIXED29BIT.		
M2 Template	**M2 Description**	
TPS_SYST	Declares which communication addressing mode is supported.	
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpConnection.addressingFormat		
Mapping Rule		**Mapping Type**
Create container if addressingFormat is set to "extended".		full

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu/CanTpNTa	
BSW Parameter		**BSW Type**
CanTpNTa		EcucIntegerParamDef
BSW Description		
This parameter contains the transport protocol target address value.		
M2 Template	**M2 Description**	
TPS_SYST	An ECUs TP address on the referenced channel. This represents the diagnostic Address.	
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpAddress.tpAddress		
Mapping Rule		**Mapping Type**
The CanTPConnection contains a reference to the SDU and a relation to the Tp Node that contains the TpAddress.		full

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu	
BSW Parameter		**BSW Type**
CanTpNar		EcucFloatParamDef
BSW Description		
Value in seconds of the N_Ar timeout. N_Ar is the time for transmission of a CAN frame (any N_PDU) on the receiver side.		
M2 Template	**M2 Description**	
TPS_SYST	This attribute states the timeout between the PDU transmit request of the Transport Layer to the FlexRay Interface and the corresponding confirmation of the FlexRay Interface on the receiver side (for FC or AF). Specified in seconds.	
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpNode.timeoutAr		

Mapping Rule	Mapping Type
1:1 mapping	full

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu	
BSW Parameter		**BSW Type**
CanTpNbr		EcucFloatParamDef
BSW Description		
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.		
M2 Template	**M2 Description**	
TPS_SYST	Value in seconds of the performance requirement for (N_Br + N_Ar).	
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpConnection.timeoutBr		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu	
BSW Parameter		**BSW Type**
CanTpNcr		EcucFloatParamDef
BSW Description		
Value in seconds of the N_Cr timeout. N_Cr is the time until reception of the next Consecutive Frame N_PDU.		
M2 Template	**M2 Description**	
TPS_SYST	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.	
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpConnection.timeoutCr		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu	
BSW Parameter		**BSW Type**
CanTpRxAddressingFormat		EcucEnumerationParamDef
BSW Description		
Declares which communication addressing mode is supported for this RxNSdu. Definition of Enumeration values: CanTpStandard to use normal addressing format. CanTpExtended to use extended addressing format. CanTpMixed to use mixed 11 bit addressing format. CanTpNormalFixed to use normal fixed addressing format. CanTpMixed29Bit to use mixed 29 bit addressing format.		
M2 Template	**M2 Description**	
TPS_SYST	Declares which communication addressing mode is supported	
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpConnection.addressingFormat		
Mapping Rule		**Mapping Type**

1:1 mapping	full

BSW Module	**BSW Context**
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu
BSW Parameter	**BSW Type**
CanTpRxDl	EcucIntegerParamDef
BSW Description	
Please note that this parameter is deprecated and will be removed in a future release. Old description: Data Length Code of this RxNsdu. In case of variable message length, this value indicates the minimum data length. Depending on SF or FF N-SDU the value will be limited to 7 (6 for an extended addressing format) and 4095 respectively.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu
BSW Parameter	**BSW Type**
CanTpRxNPdu	EcucParamConfContainerDef
BSW Description	
Used for grouping of the ID of a PDU and the Reference to a PDU.	
M2 Template	**M2 Description**
TPS_SYST	Reference to an Data NPdu.
M2 Parameter	
SystemTemplate::TransportProtocols::CanTpConnection.dataPdu	
Mapping Rule	**Mapping Type**
Create container if the CanTpConnection contains a reference to a DataNpdu that is received by the regarded ECU.	full

BSW Module	**BSW Context**
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu/CanTpRxNPdu
BSW Parameter	**BSW Type**
CanTpRxNPduId	EcucIntegerParamDef
BSW Description	
The N-PDU identifier attached to the RxNsdu is identified by CanTpRxNSduId. Each RxNsdu identifier is linked to only one SF/FF/CF N-PDU identifier. Nevertheless, in the case of extended or mixed addressing format, the same N-PDU identifier can be used for several N-SDU identifiers. The distinction is made by the N_TA or N_AE value (first data byte of SF or FF frames).	
M2 Template	**M2 Description**
M2 Parameter	

Mapping Rule	Mapping Type
	local

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu/CanTpRxNPdu	
BSW Parameter		**BSW Type**
CanTpRxNPduRef		EcucReferenceDef
BSW Description		
Reference to a Pdu in the COM-Stack.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu	
BSW Parameter		**BSW Type**
CanTpRxNSduId		EcucIntegerParamDef
BSW Description		
Unique identifier user by the upper layer to call CanTp_CancelReceive, CanTp_ChangeParameter and CanTp_ReadParameter.		
M2 Template	**M2 Description**	
TPS_SYST	To describe a frames identifier on the communication system, usualy with a fixed identifierValue.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.identifier		
Mapping Rule		**Mapping Type**
Id described by the CanId in the FrameTriggering.		full

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu	
BSW Parameter		**BSW Type**
CanTpRxNSduRef		EcucReferenceDef
BSW Description		
Reference to a Pdu in the COM-Stack.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu	
BSW Parameter		**BSW Type**
CanTpRxPaddingActivation		EcucEnumerationParamDef
BSW Description		

Defines if the receive frame uses padding or not. Definition of enumeration values: CanTpOn: The N-PDU received uses padding for SF, FC and the last CF. (N-PDU length is always 8 bytes) CanTpOff: The N-PDU received does not use padding for SF, CF and the last CF. (N-PDU length is dynamic)

M2 Template	M2 Description	
TPS_SYST	This specifies wheter or not Sfs, FCs and the last CF shall be padded to 8 bytes length in case it contains less payload.	
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpConnection.paddingActivation		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu	
BSW Parameter		**BSW Type**
CanTpRxTaType		EcucEnumerationParamDef
BSW Description		
Declares the communication type of this Rx N-SDU.		
M2 Template	**M2 Description**	
TPS_SYST	Network Target Address type.	
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpConnection.taType		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu	
BSW Parameter		**BSW Type**
CanTpRxWftMax		EcucIntegerParamDef
BSW Description		
This parameter indicates how many Flow Control wait N-PDUs can be consecutively transmitted by the receiver. It is local to the node and is not transmitted inside the FC protocol data unit. CanTpRxWftMax is used to avoid sender nodes being potentially hooked-up in case of a temporarily reception inability on the part of the receiver nodes, whereby the sender could be waiting continuously.		
M2 Template	**M2 Description**	
TPS_SYST	This attribute defines the maximum number of flow control PDUs that can be consecutively be transmitted by a receiver.	
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpNode.maxFcWait		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu

BSW Parameter		BSW Type
CanTpSTmin		EcucFloatParamDef
BSW Description		
Sets the duration of the minimum time the CanTp sender shall wait between the transmissions of two CF N-PDUs. For further details on this parameter value see ISO 15765-2 specification.		
M2 Template	**M2 Description**	
TPS_SYST	This parameter defines the minimum amount of time (in seconds) between two succeeding CFs.	
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpNode.stMin		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu	
BSW Parameter		**BSW Type**
CanTpTxFcNPdu		EcucParamConfContainerDef
BSW Description		
Used for grouping of the ID of a PDU and the Reference to a PDU.		
M2 Template	**M2 Description**	
TPS_SYST	Reference to the Flow Control NPdu.	
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpConnection.flowControlPdu		
Mapping Rule		**Mapping Type**
Create container if the CanTpConnection contains a reference to a FlowControl NPdu that is received by the regarded ECU.		full

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu/CanTpTxFcNPdu	
BSW Parameter		**BSW Type**
CanTpTxFcNPduConfirmationPduId		EcucIntegerParamDef
BSW Description		
Handle Id to be used by the CanIf to confirm the transmission of the CanTpTxFcNPdu to the CanIf module.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpRxNSdu/CanTpTxFcNPdu	
BSW Parameter		**BSW Type**
CanTpTxFcNPduRef		EcucReferenceDef
BSW Description		
Reference to a Pdu in the COM-Stack.		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**	
CanTp	CanTp/CanTpConfig/CanTpChannel	
BSW Parameter		**BSW Type**
CanTpTxNSdu		EcucParamConfContainerDef
BSW Description		
The following parameters needs to be configured for each CAN N-SDU that the CanTp module transmits via the CanTpChannel.		
M2 Template	**M2 Description**	
TPS_SYST	Reference to the IPdu that is segmented by the Transport Protocol.	
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpConnection.tpSdu		
Mapping Rule		**Mapping Type**
Create container for each existing CanTpConnection that contains a reference to an N-SDU that is transmitted.		full

BSW Module	**BSW Context**	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu	
BSW Parameter		**BSW Type**
CanTpNAe		EcucParamConfContainerDef
BSW Description		
This container is required for each RxNSdu and TxNSdu with AddressingFormat CANTP_MIXED or CANTP_MIXED29BIT.		
M2 Template	**M2 Description**	
TPS_SYST	Declares which communication addressing mode is supported.	
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpConnection.addressingFormat		
Mapping Rule		**Mapping Type**
Create container if addressingFormat is set to "mixed".		full

BSW Module	**BSW Context**	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu/CanTpNAe	
BSW Parameter		**BSW Type**
CanTpNAe		EcucIntegerParamDef
BSW Description		
This parameter contains the transport protocol address extension value.		
M2 Template	**M2 Description**	
TPS_SYST	An ECUs TP address on the referenced channel. This represents the diagnostic Address.	
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpAddress.tpAddressExtensionValue		
Mapping Rule		**Mapping Type**
The CanTPConnection contains a reference to the SDU and a relation to the Tp Node that contains the TpAddressExtension.		full

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu	
BSW Parameter		**BSW Type**
CanTpNSa		EcucParamConfContainerDef
BSW Description		
This container is required for each RxNSdu and TxNSdu with RxTaType CANTP_PHYSICAL and CanTpAddressingFormat CANTP_EXTENDED. When DynIdSupport is enabled, this container is also required for each TxNSdu with AddressingFormat CANTP_NORMALFIXED or CANTP_MIXED29BIT. When DynIdSupport is enabled and GenericConnectionSupport is not enabled, this container is also required for each RxNSdu with AddressingFormat CANTP_NORMALFIXED or CANTP_MIXED29BIT.		
M2 Template	**M2 Description**	
TPS_SYST	Declares which communication addressing mode is supported.	
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpConnection.addressingFormat		
Mapping Rule		**Mapping Type**
Create container if addressingFormat is set to "extended".		full

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu/CanTpNSa	
BSW Parameter		**BSW Type**
CanTpNSa		EcucIntegerParamDef
BSW Description		
This parameter contains the transport protocol source address value.		
M2 Template	**M2 Description**	
TPS_SYST	An ECUs TP address on the referenced channel. This represents the diagnostic Address.	
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpAddress.tpAddress		
Mapping Rule		**Mapping Type**
The CanTPConnection contains a reference to the SDU and a relation to the Tp Node that contains the TpAddress.		full

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu	
BSW Parameter		**BSW Type**
CanTpNTa		EcucParamConfContainerDef
BSW Description		
This container is required for each RxNSdu and TxNSdu with AddressingFormat CANTP_EXTENDED. When DynIdSupport is enabled, this container is also required for each RxNSdu with AddressingFormat CANTP_NORMALFIXED or CANTP_MIXED29BIT. When DynIdSupport is enabled and GenericConnectionSupport is not enabled, this container is also required for each TxNSdu with AddressingFormat CANTP_NORMALFIXED or CANTP_MIXED29BIT.		
M2 Template	**M2 Description**	
TPS_SYST	Declares which communication addressing mode is supported.	
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpConnection.addressingFormat		
Mapping Rule		**Mapping Type**
Create container if addressingFormat is set to "extended".		full

BSW Module	BSW Context

CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu/CanTpNTa
BSW Parameter	**BSW Type**
CanTpNTa	EcucIntegerParamDef
BSW Description	
This parameter contains the transport protocol target address value.	
M2 Template	**M2 Description**
TPS_SYST	An ECUs TP address on the referenced channel. This represents the diagnostic Address.
M2 Parameter	
SystemTemplate::TransportProtocols::CanTpAddress.tpAddress	
Mapping Rule	**Mapping Type**
The CanTPConnection contains a reference to the SDU and a relation to the Tp Node that contains the TpAddress.	full

BSW Module	**BSW Context**
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu
BSW Parameter	**BSW Type**
CanTpNas	EcucFloatParamDef
BSW Description	
Value in second of the N_As timeout. N_As is the time for transmission of a CAN frame (any N_PDU) on the part of the sender.	
M2 Template	**M2 Description**
TPS_SYST	Timeout between the PDU transmit request for the first PDU of the group used in the current connection of the Transport Layer to the Can Interface and the corresponding confirmation of the Can Interface on the sender side
M2 Parameter	
SystemTemplate::TransportProtocols::CanTpNode.timeoutAs	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu
BSW Parameter	**BSW Type**
CanTpNbs	EcucFloatParamDef
BSW Description	
Value in seconds of the N_Bs timeout. N_Bs is the time of transmission until reception of the next Flow Control N_PDU.	
M2 Template	**M2 Description**
TPS_SYST	This parameter defines the timout for waiting for an FC or AF on the sender side in an 1:1 connection. Specified in seconds.
M2 Parameter	
SystemTemplate::TransportProtocols::CanTpConnection.timeoutBs	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu
BSW Parameter	**BSW Type**
CanTpNcs	EcucFloatParamDef
BSW Description	

Value in seconds of the performance requirement of (N_Cs + N_As). N_Cs is the time which elapses between the transmit request of a CF N-PDU until the transmit request of the next CF N-PDU.	
M2 Template	**M2 Description**
TPS_SYST	timeoutCs is the time which elapses between the transmit request of a CF N-PDU until the transmit request of the next CF N-PDU.
M2 Parameter	
SystemTemplate::TransportProtocols::CanTpConnection.timeoutCs	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu
BSW Parameter	**BSW Type**
CanTpRxFcNPdu	EcucParamConfContainerDef
BSW Description	
Used for grouping of the ID of a PDU and the Reference to a PDU.	
M2 Template	**M2 Description**
TPS_SYST	Reference to the Flow Control NPdu.
M2 Parameter	
SystemTemplate::TransportProtocols::CanTpConnection.flowControlPdu	
Mapping Rule	**Mapping Type**
Create container if the CanTpConnection contains a reference to a FlowControl NPdu that is received by the regarded ECU.	full

BSW Module	**BSW Context**
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu/CanTpRxFcNPdu
BSW Parameter	**BSW Type**
CanTpRxFcNPduId	EcucIntegerParamDef
BSW Description	
N-PDU identifier attached to the FC N-PDU of this TxNsdu identified by CanTpTxNSduId. Each TxNsdu identifier is linked to one Rx FC N-PDU identifier only. However, in the case of extended addressing format, the same FC N-PDU identifier can be used for several N-SDU identifiers. The distinction is made by means of the N_TA value (first data byte of FC frames).	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu/CanTpRxFcNPdu
BSW Parameter	**BSW Type**
CanTpRxFcNPduRef	EcucReferenceDef
BSW Description	
Reference to a Pdu in the COM-Stack.	
M2 Template	**M2 Description**
M2 Parameter	

Mapping Rule	Mapping Type
	local

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu	
BSW Parameter		BSW Type
CanTpTc		EcucBooleanParamDef
BSW Description		
Switch for enabling Transmit Cancellation and Receive Cancellation.		
M2 Template	M2 Description	
TPS_SYST	With this switch Tx and Rx Cancellation can be turned on or off.	
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpConnection.cancellation		
Mapping Rule		Mapping Type
1:1 mapping		full

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu	
BSW Parameter		BSW Type
CanTpTxAddressingFormat		EcucEnumerationParamDef
BSW Description		
Declares which communication addressing format is supported for this TxNSdu. Definition of Enumeration values: CanTpStandard to use normal addressing format. CanTpExtended to use extended addressing format. CanTpMixed to use mixed 11 bit addressing format. CanTpNormalFixed to use normal fixed addressing format. CanTpMixed29Bit to use mixed 29 bit addressing format.		
M2 Template	M2 Description	
TPS_SYST	Declares which communication addressing mode is supported.	
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpConnection.addressingFormat		
Mapping Rule		Mapping Type
1:1 mapping		full

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu	
BSW Parameter		BSW Type
CanTpTxDl		EcucIntegerParamDef
BSW Description		
Please note that this parameter is deprecated and will be removed in a future release. Old description: Data Length Code of this TxNsdu. In case of variable length message, this value indicates the minimum data length.		
M2 Template	M2 Description	
M2 Parameter		
Mapping Rule		Mapping Type

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu	
BSW Parameter		**BSW Type**
CanTpTxNPdu		EcucParamConfContainerDef
BSW Description		
Used for grouping of the ID of a PDU and the Reference to a PDU.		
M2 Template	**M2 Description**	
TPS_SYST	Reference to an Data NPdu.	
M2 Parameter		
SystemTemplate::TransportProtocols::CanTpConnection.dataPdu		
Mapping Rule		**Mapping Type**
Create container if the CanTpConnection contains a reference to a DataNpdu that is received by the regarded ECU.		full

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu/CanTpTxNPdu	
BSW Parameter		**BSW Type**
CanTpTxNPduConfirmationPduId		EcucIntegerParamDef
BSW Description		
Handle Id to be used by the CanIf to confirm the transmission of the CanTpTxNPdu to the CanIf module.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu/CanTpTxNPdu	
BSW Parameter		**BSW Type**
CanTpTxNPduRef		EcucReferenceDef
BSW Description		
Reference to a Pdu in the COM-Stack.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu	
BSW Parameter		**BSW Type**
CanTpTxNSduId		EcucIntegerParamDef
BSW Description		

Unique identifier to a structure that contains all useful information to process the transmission of a TxNsdu.	
M2 Template	**M2 Description**
TPS_SYST	To describe a frames identifier on the communication system, usualy with a fixed identifierValue.
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanCommunication::CanFrameTriggering.identifier	
Mapping Rule	**Mapping Type**
Id described by the CanId in the FrameTriggering.	full

BSW Module	**BSW Context**
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu
BSW Parameter	**BSW Type**
CanTpTxNSduRef	EcucReferenceDef
BSW Description	
Reference to a Pdu in the COM-Stack.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu
BSW Parameter	**BSW Type**
CanTpTxPaddingActivation	EcucEnumerationParamDef
BSW Description	
Defines if the transmit frame use padding or not. Definition of Enumeration values: CanTpOn The transmit N-PDU uses padding for SF, FC and the last CF. (N-PDU length is always 8 bytes) CanTpOff The transmit N-PDU does not use padding for SF, CF and the last CF. (N-PDU length is dynamic)	
M2 Template	**M2 Description**
TPS_SYST	This specifies wheter or not Sfs, FCs and the last CF shall be padded to 8 bytes length in case it contains less payload.
M2 Parameter	
SystemTemplate::TransportProtocols::CanTpConnection.paddingActivation	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
CanTp	CanTp/CanTpConfig/CanTpChannel/CanTpTxNSdu
BSW Parameter	**BSW Type**
CanTpTxTaType	EcucEnumerationParamDef
BSW Description	

Declares the communication type of this TxNsdu. Enumeration values: CanTpPhysical. Used for 1:1 communication. CanTpFunctional. Used for 1:n communication.	
M2 Template	**M2 Description**
TPS_SYST	Network Target Address type.
M2 Parameter	
SystemTemplate::TransportProtocols::CanTpConnection.taType	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
CanTp	CanTp/CanTpConfig
BSW Parameter	**BSW Type**
CanTpMainFunctionPeriod	EcucFloatParamDef
BSW Description	
Allow to configure the time for the MainFunction (as float in seconds). The CanTpMainFunctionPeriod should be assigned a value which is optimal regarding all of the timers configured for CanTp in TX and RX data transfer i.e. the differences from the configured timing should be as small as possible. Please note: This period shall be the same as call cycle time of the periodic task were CanTp Main function is called.	
M2 Template	**M2 Description**
TPS_SYST	The period between successive calls to the Main Function of the ASR TP. Specified in seconds.
M2 Parameter	
SystemTemplate::TransportProtocols::CanTpEcu.cycleTimeMainFunction	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
CanTp	CanTp/CanTpConfig
BSW Parameter	**BSW Type**
CanTpMaxChannelCnt	EcucIntegerParamDef
BSW Description	
Maximum number of channels. This parameter is needed only in case of post-build loadable implementation using static memory allocation.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**
CanTp	CanTp
BSW Parameter	**BSW Type**
CanTpGeneral	EcucParamConfContainerDef
BSW Description	
This container contains the general configuration parameters of the CanTp module.	
M2 Template	**M2 Description**

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**	
CanTp	CanTp/CanTpGeneral	
BSW Parameter		**BSW Type**
CanTpChangeParameterApi		EcucBooleanParamDef
BSW Description		
This parameter, if set to true, enables the CanTp_ChangeParameterRequest Api for this Module.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
CanTp	CanTp/CanTpGeneral	
BSW Parameter		**BSW Type**
CanTpDevErrorDetect		EcucBooleanParamDef
BSW Description		
Switches the Development Error Detection and Notification ON or OFF		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
CanTp	CanTp/CanTpGeneral	
BSW Parameter		**BSW Type**
CanTpDynIdSupport		EcucBooleanParamDef
BSW Description		
Enable support for dynamic ID handling via N-PDU MetaData.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
CanTp	CanTp/CanTpGeneral	
BSW Parameter		**BSW Type**
CanTpGenericConnectionSupport		EcucBooleanParamDef

BSW Description		
Enable support for the handling of generic connections using N-SDUs with MetaData. Requires CanTpDynIdSupport.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanTp	CanTp/CanTpGeneral	
BSW Parameter		**BSW Type**
CanTpPaddingByte		EcucIntegerParamDef
BSW Description		
Used for the initialization of unused bytes with a certain value		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanTp	CanTp/CanTpGeneral	
BSW Parameter		**BSW Type**
CanTpReadParameterApi		EcucBooleanParamDef
BSW Description		
This parameter, if set to true, enables the CanTp_ReadParameterApi for this module.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
CanTp	CanTp/CanTpGeneral	
BSW Parameter		**BSW Type**
CanTpVersionInfoApi		EcucBooleanParamDef
BSW Description		
The function CanTp_GetVersionInfo is configurable (On/Off) by this configuration parameter.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

D.3 J1939

D.3.1 J1939Tp Mapping

BSW Module	**BSW Context**	
J1939Tp	J1939Tp	
BSW Parameter		**BSW Type**
J1939TpConfiguration		EcucParamConfContainerDef
BSW Description		
This container contains the configuration parameters and sub containers of the J1939Tp module that define the communication paths. This container is a MultipleConfigurationContainer, i.e. this container and its sub-containers exist once per configuration set.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
J1939Tp	J1939Tp/J1939TpConfiguration	
BSW Parameter		**BSW Type**
J1939TpRxChannel		EcucParamConfContainerDef
BSW Description		
This container describes a reception channel of the J1939Tp module. A channel referencing N-PDUs without MetaData is used for all N-SDUs that share the same source address (SA) and the same destination address (BAM: DA = 0xFF, CMDT: DA != 0xFF). A channel with N-PDUs with MetaData is used for all possible source and destination addresses.		
M2 Template	**M2 Description**	
TPS_SYST	A J1939TpChannel represents an internal path for the transmission or reception of a Pdu via J1939Tp and describes the sender and the receiver of this particular connection. The J1939Tp module routes a Pdu (J1939 PGN) through the channel.	
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpConnection		
Mapping Rule		**Mapping Type**
Create container for each existing J1939TpConnection that is used to transmit a NSdu.		full

BSW Module	**BSW Context**	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel	
BSW Parameter		**BSW Type**
J1939TpRxCancellationSupport		EcucBooleanParamDef
BSW Description		
Enable receive cancellation using the API J1939Tp_CancelReceive() for this channel.		
M2 Template	**M2 Description**	
TPS_SYST	Enable support for Tx/Rx cancellation.	
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpConnection.cancellation		
Mapping Rule		**Mapping Type**

Please note that in the System Template the cancellation support is defined per J1939TpConnection. All J1939TpConnections in an ECU shall have the same value.	full

BSW Module	**BSW Context**		
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel		
BSW Parameter		**BSW Type**	
J1939TpRxCmNPdu		EcucParamConfContainerDef	
BSW Description			
This N-PDU represents the TP.CM frame of a J1939 transport protocol session. TP.CM is used both by BAM and CMDT to initialize the connection. For CMDT, it is also used to abort the connection.			
M2 Template	**M2 Description**		
TPS_SYST	Reference to the Flow Control NPdus that are used in the CMDT (Connection Mode Data Transfer) for TP.CM in both directions. BAM uses one TP.CM (Transport Protocol Command).		
M2 Parameter			
SystemTemplate::TransportProtocols::J1939TpConnection.flowControlPdu			
Mapping Rule			**Mapping Type**
Information can be derived from a received directlNPdu that is referenced by the J1939TpConnection.			full

BSW Module	**BSW Context**		
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel/J1939TpRxCmNPdu		
BSW Parameter		**BSW Type**	
J1939TpRxCmNPduId		EcucIntegerParamDef	
BSW Description			
The N-PDU identifier used for communication with CanIf.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel/J1939TpRxCmNPdu		
BSW Parameter		**BSW Type**	
J1939TpRxCmNPduRef		EcucReferenceDef	
BSW Description			
Reference to the Pdu object representing the N-PDU.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel

BSW Parameter			BSW Type	
J1939TpRxDa			EcucIntegerParamDef	
BSW Description				
Destination address (DA) of this channel. This parameter is only required for channels with fixed DA which use N-PDUs with MetaData containing the DA.				
M2 Template	**M2 Description**			
M2 Parameter				
Mapping Rule				**Mapping Type**
				local

BSW Module	BSW Context			
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel			
BSW Parameter			**BSW Type**	
J1939TpRxDtNPdu			EcucParamConfContainerDef	
BSW Description				
This N-PDU represents the TP.DT frame of a J1939 transport protocol session. TP.DT is used both by BAM and CMDT to transfer the contents of an N-SDU.				
M2 Template	**M2 Description**			
TPS_SYST	There are two transport protocols defined for J1939: BAM (Broadcast Announce Message), which is a broadcast protocol, and CMDT (Connection Mode Data Transfer), which is a point-to-point protocol with flow control.			
M2 Parameter				
SystemTemplate::TransportProtocols::J1939TpConnection.dataPdu				
Mapping Rule				**Mapping Type**
Information can be derived from a received NPdu that is referenced by the J1939 TpConnection.				full

BSW Module	BSW Context			
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel/J1939TpRxDtNPdu			
BSW Parameter			**BSW Type**	
J1939TpRxDtNPduId			EcucIntegerParamDef	
BSW Description				
The N-PDU identifier used for communication with CanIf.				
M2 Template	**M2 Description**			
M2 Parameter				
Mapping Rule				**Mapping Type**
				local

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel/J1939TpRxDtNPdu	
BSW Parameter		**BSW Type**
J1939TpRxDtNPduRef		EcucReferenceDef
BSW Description		
Reference to the Pdu object representing the N-PDU.		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel	
BSW Parameter		**BSW Type**
J1939TpRxDynamicBlockCalculation		EcucBooleanParamDef
BSW Description		
Enable dynamic calculation of "number of packets that can be sent" value in TP.CM_CTS, based on the size of buffers in upper layers reported via StartOfReception and PduR_J1939TpCopyRxData.		
M2 Template	**M2 Description**	
TPS_SYST	Enable support for dynamic block size calculation.	
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpConnection.dynamicBs		
Mapping Rule		**Mapping Type**
Please note that in the System Template the dynamic block size calculation support is defined per J1939TpConnection. All J1939TpConnections in an EC U shall have the same value.		full

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel	
BSW Parameter		**BSW Type**
J1939TpRxDynamicBufferRatio		EcucIntegerParamDef
BSW Description		
Percentage of available buffer that shall be used for retry. This parameter is only applicable when "J1939TpRxRetrySupport" and "J1939TpRxDynamicBlockCalculation" are enabled.		
M2 Template	**M2 Description**	
TPS_SYST	Defines usage of available data for dynamic block size calculation when protocol retry is enabled.	
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpConnection.bufferRatio		
Mapping Rule		**Mapping Type**
Please note that in the System Template this attribute is defined per J1939Tp Connection. All J1939TpConnections in an ECU shall have the same value.		full

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel	
BSW Parameter		**BSW Type**
J1939TpRxPacketsPerBlock		EcucIntegerParamDef
BSW Description		
Number of TP.DT frames the receiving J1939Tp module allows the sender to send before waiting for another TP.CM_CTS. This parameter is transmitted in the TP.CM_CTS frame, and is thus only relevant for reception of messages via CMDT. When J1939TpRxDynamicBlockCalculation is enabled, this parameter specifies a maximum for the calculated value. For further details on this parameter value see SAE J1939/21.		
M2 Template	**M2 Description**	
TPS_SYST	Set maximum block size (number of packets in TP.CM_CTS).	
M2 Parameter		

SystemTemplate::TransportProtocols::J1939TpConnection.maxBs	
Mapping Rule	**Mapping Type**
Please note that in the System Template the maximum block size is defined per J1939TpConnection. All J1939TpConnections in an ECU shall have the same value.	full

BSW Module	**BSW Context**	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel	
BSW Parameter		**BSW Type**
J1939TpRxPg		EcucParamConfContainerDef
BSW Description		
Parameter group received by the J1939 transport layer.		
M2 Template	**M2 Description**	
TPS_SYST	A J1939TpPg represents one J1939 message (parameter group, PG) identified by the PGN (parameter group number) that can be received or transmitted via J1939Tp.	
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpPg		
Mapping Rule		**Mapping Type**
Create container for each Rx J1939TpPg that is available in the Ecu Extract.		full

BSW Module	**BSW Context**	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel/J1939TpRxPg	
BSW Parameter		**BSW Type**
J1939TpRxDirectNPdu		EcucParamConfContainerDef
BSW Description		
This N-PDU represents the short frame that is used for a dynamic length PGN when it has a length of less that 8 bytes. Please note: This sub container is only necessary when J1939TpRxPgDynLength is TRUE.		
M2 Template	**M2 Description**	
TPS_SYST	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.	
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpPg.directPdu		
Mapping Rule		**Mapping Type**
Information can be derived from a received directINPdu that is referenced by the J1939TpPg.		full

BSW Module	**BSW Context**	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel/J1939TpRxPg/J1939TpRxDirectNPdu	
BSW Parameter		**BSW Type**
J1939TpRxDirectNPduId		EcucIntegerParamDef
BSW Description		
The N-PDU identifier used for communication with CanIf.		
M2 Template	**M2 Description**	
M2 Parameter		

Mapping Rule	Mapping Type
	local

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel/J1939TpRxPg/J1939TpRxDirectNPdu	
BSW Parameter		**BSW Type**
J1939TpRxDirectNPduRef		EcucReferenceDef
BSW Description		
Reference to the Pdu object representing the N-PDU.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel/J1939TpRxPg	
BSW Parameter		**BSW Type**
J1939TpRxNSdu		EcucParamConfContainerDef
BSW Description		
This container describes the parameters that are relevant for the reception of a specific N-SDU.		
M2 Template	**M2 Description**	
TPS_SYST	Reference to IPdus that are segmented by the Transport Protocol.	
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpConnection.tpSdu		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel/J1939TpRxPg/J1939TpRxNSdu	
BSW Parameter		**BSW Type**
J1939TpRxNSduId		EcucIntegerParamDef
BSW Description		
This is a unique identifier for a received N-SDU. This Id is used in the CancelReceive and ChangeParameter API call.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel/J1939TpRxPg/J1939TpRxNSdu	
BSW Parameter		**BSW Type**

J1939TpRxNSduRef		EcucReferenceDef
BSW Description		
Reference to the Pdu object representing the N-SDU.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel/J1939TpRxPg	
BSW Parameter		**BSW Type**
J1939TpRxPgDynLength		EcucBooleanParamDef
BSW Description		
This flag is set to TRUE when the N-SDU refers to a PGN with variable length. Please note: When this attribute is TRUE, the sub container J1939TpRxDirectNPdu is required.		
M2 Template	**M2 Description**	
TPS_SYST	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).	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::SystemSignal.dynamicLength		
Mapping Rule		**Mapping Type**
If a tpSdu that is referenced by the J1939TpPg contains a dynamicLengthSignal than set this parameter to true.		full

BSW Module	**BSW Context**	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel/J1939TpRxPg	
BSW Parameter		**BSW Type**
J1939TpRxPgPGN		EcucIntegerParamDef
BSW Description		
PGN of the referenced N-SDUs.		
M2 Template	**M2 Description**	
TPS_SYST	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 CanFrameTriggering with an identifier.	
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpPg.pgn		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel	
BSW Parameter		**BSW Type**
J1939TpRxProtocolType		EcucEnumerationParamDef
BSW Description		
Protocol type used by this channel. This parameter is only required for channels with fixed destination address.		

M2 Template	M2 Description	
TPS_SYST	BAM (Broadcast Announce Message) is a broadcast protocol. If this attribute is set to true broadcast is used. Since address FF is the only broadcast address, there's no reason to configure it.	
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpConnection.broadcast		
Mapping Rule		**Mapping Type**
If the braodcast attribute is set to true than set this parameter to J1939TP_PROTOCOL_BAM		full

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel	
BSW Parameter		**BSW Type**
J1939TpRxRetrySupport		EcucBooleanParamDef
BSW Description		
Enable support for triggering repetition of failed transmission using TP.CM_CTS with a packet number that has already been sent. Retransmission is triggered when a sequence number is missing or a timeout occurs during reception.		
M2 Template	**M2 Description**	
TPS_SYST	Enable support for protocol retry.	
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpConnection.retry		
Mapping Rule		**Mapping Type**
Please note that in the System Template the retry support is defined per J1939TpConnection. All J1939TpConnections in an ECU shall have the same value.		full

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel	
BSW Parameter		**BSW Type**
J1939TpRxSa		EcucIntegerParamDef
BSW Description		
Source address (SA) of this channel. This parameter is only required for channels with fixed SA which use N-PDUs with MetaData.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel	
BSW Parameter		**BSW Type**
J1939TpTxFcNPdu		EcucParamConfContainerDef
BSW Description		
This N-PDU represents the TP.CM frame that is used in reverse direction for a J1939 transport protocol session using the CMDT protocol type. TP.CM in reverse direction is used for intermediate and final acknowledgement of received data and to abort the connection. Please note: This sub container is only required when J1939TpRxProtocolType is J1939TP_PROTOCOL_CMDT or when it is not configured at all.		

M2 Template	M2 Description	
TPS_SYST	Reference to the Flow Control NPdus that are used in the CMDT (Connection Mode Data Transfer) for TP.CM in both directions. BAM uses one TP.CM (Transport Protocol Command).	
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpConnection.flowControlPdu		
Mapping Rule		**Mapping Type**
Information can be derived from a received FlowControlNPdu that is referenced by the J1939TpConnection.		full

BSW Module	BSW Context	BSW Type
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel/J1939TpTxFcNPdu	
BSW Parameter		**BSW Type**
J1939TpTxFcNPduRef		EcucReferenceDef
BSW Description		
Reference to the Pdu object representing the N-PDU. Please note: When two channels have identical but exchanged source and destination addresses, the Pdu referenced by this parameter is shared with J1939TpTxCmNPduRef of the corresponding J1939TpTxChannel.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	BSW Type
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpRxChannel/J1939TpTxFcNPdu	
BSW Parameter		**BSW Type**
J1939TpTxFcNPduTxConfId		EcucIntegerParamDef
BSW Description		
The N-PDU identifier used for Tx confirmation from CanIf.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	BSW Type
J1939Tp	J1939Tp/J1939TpConfiguration	
BSW Parameter		**BSW Type**
J1939TpTxChannel		EcucParamConfContainerDef
BSW Description		
This container describes a transmission channel of the J1939Tp module. A channel referencing N-PDUs without MetaData is used for all N-SDUs that share the same source address (SA) and the same destination address (BAM: DA = 0xFF, CMDT: DA != 0xFF). A channel with N-PDUs with MetaData is used for all possible source and destination addresses.		
M2 Template	**M2 Description**	

TPS_SYST	A J1939TpChannel represents an internal path for the transmission or reception of a Pdu via J1939Tp and describes the sender and the receiver of this particular connection. The J1939Tp module routes a Pdu (J1939 PGN) through the channel.	
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpConnection		
Mapping Rule		**Mapping Type**
Create container for each existing J1939TpConnection that is used to transmit a NSdu.		full

BSW Module	**BSW Context**	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel	
BSW Parameter		**BSW Type**
J1939TpRxFcNPdu		EcucParamConfContainerDef
BSW Description		
This N-PDU represents the TP.CM frame that is used in reverse direction for a J1939 transport protocol session using the CMDT protocol type. TP.CM in reverse direction is used for intermediate and final acknowledgement of received data and to abort the connection. Please note: This sub container is only required when J1939TpRxProtocolType is J1939TP_PROTOCOL_CMDT or when it is not configured at all.		
M2 Template	**M2 Description**	
TPS_SYST	Reference to the Flow Control NPdus that are used in the CMDT (Connection Mode Data Transfer) for TP.CM in both directions. BAM uses one TP.CM (Transport Protocol Command).	
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpConnection.flowControlPdu		
Mapping Rule		**Mapping Type**
Information can be derived from a transmitted FlowControlNPdu that is referenced by the J1939TpConnection.		full

BSW Module	**BSW Context**	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel/J1939TpRxFcNPdu	
BSW Parameter		**BSW Type**
J1939TpRxFcNPduId		EcucIntegerParamDef
BSW Description		
The N-PDU identifier used for communication with CanIf.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel/J1939TpRxFcNPdu	
BSW Parameter		**BSW Type**
J1939TpRxFcNPduRef		EcucReferenceDef
BSW Description		

Reference to the Pdu object representing the N-PDU. Please note: When two channels have identical but exchanged source and destination addresses, the Pdu referenced by this parameter is shared with J1939TpRxCmNPduRef of the corresponding J1939TpRxChannel.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel	
BSW Parameter		**BSW Type**
J1939TpTxCancellationSupport		EcucBooleanParamDef
BSW Description		
Enable transmit cancellation using the API J1939Tp_CancelTransmit() for this channel.		
M2 Template	**M2 Description**	
TPS_SYST	Enable support for Tx/Rx cancellation.	
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpConnection.cancellation		
Mapping Rule		**Mapping Type**
Please note that in the System Template the cancellation support is defined per J1939TpConnection. All J1939TpConnections in an ECU shall have the same value.		full

BSW Module	**BSW Context**	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel	
BSW Parameter		**BSW Type**
J1939TpTxCmNPdu		EcucParamConfContainerDef
BSW Description		
This N-PDU represents the TP.CM frame of a J1939 transport protocol session. TP.CM is used both by BAM and CMDT to initialize the connection. For CMDT, it is also used to abort the connection.		
M2 Template	**M2 Description**	
TPS_SYST	Reference to the Flow Control NPdus that are used in the CMDT (Connection Mode Data Transfer) for TP.CM in both directions. BAM uses one TP.CM (Transport Protocol Command).	
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpConnection.flowControlPdu		
Mapping Rule		**Mapping Type**
Information can be derived from a transmitted FlowControlNPdu that is referenced by the J1939TpConnection.		full

BSW Module	**BSW Context**	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel/J1939TpTxCmNPdu	
BSW Parameter		**BSW Type**
J1939TpTxCmNPduRef		EcucReferenceDef
BSW Description		
Reference to the Pdu object representing the N-PDU.		
M2 Template	**M2 Description**	

M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel/J1939TpTxCmNPdu	
BSW Parameter		**BSW Type**
J1939TpTxCmNPduTxConfId		EcucIntegerParamDef
BSW Description		
The N-PDU identifier used for Tx confirmation from CanIf.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel	
BSW Parameter		**BSW Type**
J1939TpTxDa		EcucIntegerParamDef
BSW Description		
Destination address (DA) of this channel. This parameter is only required for channels with fixed DA which use N-PDUs with MetaData containing the DA.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel	
BSW Parameter		**BSW Type**
J1939TpTxDtNPdu		EcucParamConfContainerDef
BSW Description		
This N-PDU represents the TP.DT frame of a J1939 transport protocol session. TP.DT is used both by BAM and CMDT to transfer the contents of an N-SDU.		
M2 Template	**M2 Description**	
TPS_SYST	There are two transport protocols defined for J1939: BAM (Broadcast Announce Message), which is a broadcast protocol, and CMDT (Connection Mode Data Transfer), which is a point-to-point protocol with flow control.	
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpConnection.dataPdu		
Mapping Rule		**Mapping Type**
Information can be derived from a transmitted NPdu that is referenced by the J1939TpConnection.		full

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel/J1939TpTxDtNPdu	
BSW Parameter		**BSW Type**
J1939TpTxDtNPduRef		EcucReferenceDef
BSW Description		
Reference to the Pdu object representing the N-PDU.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel/J1939TpTxDtNPdu	
BSW Parameter		**BSW Type**
J1939TpTxDtNPduTxConfId		EcucIntegerParamDef
BSW Description		
The N-PDU identifier used for Tx confirmation from CanIf.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel	
BSW Parameter		**BSW Type**
J1939TpTxDynamicBlockCalculation		EcucBooleanParamDef
BSW Description		
Enable dynamic calculation of "maximum number of packets that can be sent" value in TP.CM_RTS, based on the available amount of data in upper layers reported via PduR_J1939TpCopyTxData.		
M2 Template	**M2 Description**	
TPS_SYST	Enable support for dynamic block size calculation.	
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpConnection.dynamicBs		
Mapping Rule		**Mapping Type**
Please note that in the System Template the dynamic block size calculation support is defined per J1939TpConnection. All J1939TpConnections in an EC U shall have the same value.		full

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel	
BSW Parameter		**BSW Type**
J1939TpTxMaxPacketsPerBlock		EcucIntegerParamDef
BSW Description		

Maximum number of TP.DT frames the transmitting J1939Tp module is ready to send before waiting for another TP.CM_CTS. This parameter is transmitted in the TP.CM_RTS frame, and is thus only relevant for transmission of messages via CMDT. When J1939TpTxDynamicBlockCalculation is enabled, this parameter specifies a maximum for the calculated value. For further details on this parameter value see SAE J1939/21.		
M2 Template	**M2 Description**	
TPS_SYST	Set maximum for expected block size (maximum number of packets in TP.CM_RTS).	
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpConnection.maxExpBs		
Mapping Rule		**Mapping Type**
Please note that in the System Template the maximum for expected block size is defined per J1939TpConnection. All J1939TpConnections in an ECU shall have the same value.		full

BSW Module	**BSW Context**		
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel		
BSW Parameter		**BSW Type**	
J1939TpTxPg		EcucParamConfContainerDef	
BSW Description			
Parameter group transmitted by the J1939 transport layer.			
M2 Template	**M2 Description**		
TPS_SYST	A J1939TpPg represents one J1939 message (parameter group, PG) identified by the PGN (parameter group number) that can be received or transmitted via J1939Tp.		
M2 Parameter			
SystemTemplate::TransportProtocols::J1939TpPg			
Mapping Rule			**Mapping Type**
Create container for each Tx J1939TpPg that is available in the Ecu Extract.			full

BSW Module	**BSW Context**		
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel/J1939TpTxPg		
BSW Parameter		**BSW Type**	
J1939TpTxDirectNPdu		EcucParamConfContainerDef	
BSW Description			
This N-PDU represents the short frame that is used for a dynamic length PGN when it has a length of less that 8 bytes. Please note: This sub container is only necessary when J1939TpTxPgDynLength is TRUE.			
M2 Template	**M2 Description**		
TPS_SYST	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.		
M2 Parameter			
SystemTemplate::TransportProtocols::J1939TpPg.directPdu			
Mapping Rule			**Mapping Type**
Information can be derived from a transmitted directINPdu that is referenced by the J1939TpPg.			full

BSW Module	**BSW Context**

J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel/J1939TpTxPg/J1939TpTxDirectNPdu		
BSW Parameter		**BSW Type**	
J1939TpTxDirectNPduRef		EcucReferenceDef	
BSW Description			
Reference to the Pdu object representing the N-PDU.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel/J1939TpTxPg/J1939TpTxDirectNPdu		
BSW Parameter		**BSW Type**	
J1939TpTxDirectNPduTxConfId		EcucIntegerParamDef	
BSW Description			
The N-PDU identifier used for Tx confirmation from CanIf.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel/J1939TpTxPg		
BSW Parameter		**BSW Type**	
J1939TpTxNSdu		EcucParamConfContainerDef	
BSW Description			
This container describes the parameters that are relevant for the transmission of a specific N-SDU.			
M2 Template	**M2 Description**		
TPS_SYST	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.		
M2 Parameter			
SystemTemplate::TransportProtocols::J1939TpPg.sdu			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	**BSW Context**		
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel/J1939TpTxPg/J1939TpTxNSdu		
BSW Parameter		**BSW Type**	
J1939TpTxNSduId		EcucIntegerParamDef	
BSW Description			
The N-SDU identifier used for communication with PduR.			

M2 Template	M2 Description	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context		
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel/J1939TpTxPg/J1939TpTxNSdu		
BSW Parameter		**BSW Type**	
J1939TpTxNSduRef		EcucReferenceDef	
BSW Description			
Reference to the Pdu object representing the N-SDU.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel/J1939TpTxPg		
BSW Parameter		**BSW Type**	
J1939TpTxPgDynLength		EcucBooleanParamDef	
BSW Description			
This flag is set to TRUE when the N-SDU refers to a PGN with variable length. Please note: When this attribute is TRUE, the sub container J1939TpTxDirectNPdu is required.			
M2 Template	**M2 Description**		
TPS_SYST	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).		
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreCommunication::SystemSignal.dynamicLength			
Mapping Rule			**Mapping Type**
If a tpSdu that is referenced by the J1939TpPg contains a dynamicLengthSignal than set this parameter to true.			full

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel/J1939TpTxPg	
BSW Parameter		**BSW Type**
J1939TpTxPgPGN		EcucIntegerParamDef
BSW Description		
PGN of the referenced N-SDUs.		
M2 Template	**M2 Description**	
TPS_SYST	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 CanFrameTriggering with an identifier.	
M2 Parameter		

SystemTemplate::TransportProtocols::J1939TpPg.pgn	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel
BSW Parameter	**BSW Type**
J1939TpTxProtocolType	EcucEnumerationParamDef
BSW Description	
Protocol type used by this channel. This parameter is only required for channels with fixed destination address.	
M2 Template	**M2 Description**
TPS_SYST	BAM (Broadcast Announce Message) is a broadcast protocol. If this attribute is set to true broadcast is used. Since address FF is the only broadcast address, there's no reason to configure it.
M2 Parameter	
SystemTemplate::TransportProtocols::J1939TpConnection.broadcast	
Mapping Rule	**Mapping Type**
If the braodcast attribute is set to true than set this parameter to J1939TP_PROTOCOL_BAM	full

BSW Module	**BSW Context**
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel
BSW Parameter	**BSW Type**
J1939TpTxRetrySupport	EcucBooleanParamDef
BSW Description	
Enable support for repetition of failed transmission using TP.CM_CTS with a packet number that has already been sent. Retransmission is handled via the retry feature of PduR_J1939TpCopyTxData.	
M2 Template	**M2 Description**
TPS_SYST	Enable support for protocol retry.
M2 Parameter	
SystemTemplate::TransportProtocols::J1939TpConnection.retry	
Mapping Rule	**Mapping Type**
Please note that in the System Template the retry support is defined per J1939TpConnection. All J1939TpConnections in an ECU shall have the same value.	full

BSW Module	**BSW Context**
J1939Tp	J1939Tp/J1939TpConfiguration/J1939TpTxChannel
BSW Parameter	**BSW Type**
J1939TpTxSa	EcucIntegerParamDef
BSW Description	
Source address (SA) of this channel. This parameter is only required for channels with fixed SA which use N-PDUs with MetaData.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context	
J1939Tp	J1939Tp	
BSW Parameter		**BSW Type**
J1939TpGeneral		EcucParamConfContainerDef
BSW Description		
This container describes the general configuration parameters of the J1939Tp module.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpGeneral	
BSW Parameter		**BSW Type**
J1939TpCancellationSupport		EcucBooleanParamDef
BSW Description		
Enable transmit and receive cancellation. The APIs J1939Tp_CancelTransmit() and J1939Tp_CancelReceive() will only be available when this parameter is enabled.		
M2 Template	**M2 Description**	
TPS_SYST	Enable support for Tx/Rx cancellation.	
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpConnection.cancellation		
Mapping Rule		**Mapping Type**
Please note that in the System Template the cancellation support is defined per J1939TpConnection. All J1939TpConnections in an ECU shall have the same value.		full

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpGeneral	
BSW Parameter		**BSW Type**
J1939TpDevErrorDetect		EcucBooleanParamDef
BSW Description		
Switches the Development Error Detection and Notification.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
J1939Tp	J1939Tp/J1939TpGeneral	
BSW Parameter		**BSW Type**
J1939TpDynamicBlockCalculation		EcucBooleanParamDef
BSW Description		

This parameter is deprecated and will be removed in future. Old description: Enable dynamic calculation of packets per block values in TP.CM_RTS and TP.CM_CTS, based on the size of buffers in upper layers reported via StartOfReception, CopyRxData, and CopyTxData.		
M2 Template	**M2 Description**	
TPS_SYST	Enable support for dynamic block size calculation.	
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpConnection.dynamicBs		
Mapping Rule		**Mapping Type**
Please note that in the System Template the dynamic block size calculation support is defined per J1939TpConnection. All J1939TpConnections in an EC U shall have the same value.		full

BSW Module	**BSW Context**	
J1939Tp	J1939Tp/J1939TpGeneral	
BSW Parameter		**BSW Type**
J1939TpDynamicBufferRatio		EcucIntegerParamDef
BSW Description		
This parameter is deprecated and will be removed in future. Old description: Percentage of available buffer that shall be used for retry. This parameter is only applicable when "J1939TpRetrySupport" and "J1939TpDynamicBlockCalculation" are enabled.		
M2 Template	**M2 Description**	
TPS_SYST	Defines usage of available data for dynamic block size calculation when protocol retry is enabled.	
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpConnection.bufferRatio		
Mapping Rule		**Mapping Type**
Please note that in the System Template this attribute is defined per J1939Tp Connection. All J1939TpConnections in an ECU shall have the same value.		full

BSW Module	**BSW Context**	
J1939Tp	J1939Tp/J1939TpGeneral	
BSW Parameter		**BSW Type**
J1939TpMainFunctionPeriod		EcucFloatParamDef
BSW Description		
Allow to configure the time for the MainFunction (in seconds). Please note: This configuration value shall be equal to the value in the ScheduleManager module.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
J1939Tp	J1939Tp/J1939TpGeneral	
BSW Parameter		**BSW Type**

J1939TpMaxPacketsPerBlock		EcucIntegerParamDef	
BSW Description			
This parameter is deprecated and will be removed in future. Old description: Maximum number of TP.DT frames the transmitting J1939Tp module is ready to send before waiting for another TP.CM_CTS. This parameter is transmitted in the TP.CM_RTS frame, and is thus only relevant for transmission of messages via CMDT. When J1939TpDynamicBlockCalculation is enabled, this parameter specifies a maximum for the calculated value. For further details on this parameter value see SAE J1939/21.			
M2 Template	**M2 Description**		
TPS_SYST	Set maximum for expected block size (maximum number of packets in TP.CM_RTS).		
M2 Parameter			
SystemTemplate::TransportProtocols::J1939TpConnection.maxExpBs			
Mapping Rule			**Mapping Type**
Please note that in the System Template the maximum for expected block size is defined per J1939TpConnection. All J1939TpConnections in an ECU shall have the same value.			full

BSW Module	**BSW Context**		
J1939Tp	J1939Tp/J1939TpGeneral		
BSW Parameter		**BSW Type**	
J1939TpPacketsPerBlock		EcucIntegerParamDef	
BSW Description			
This parameter is deprecated and will be removed in future. Old description: Number of TP.DT frames the receiving J1939Tp module allows the sender to send before waiting for another TP.CM_CTS. This parameter is transmitted in the TP.CM_CTS frame, and is thus only relevant for reception of messages via CMDT. When J1939TpDynamicBlockCalculation is enabled, this parameter specifies a maximum for the calculated value. For further details on this parameter value see SAE J1939/21.			
M2 Template	**M2 Description**		
TPS_SYST	Set maximum block size (number of packets in TP.CM_CTS).		
M2 Parameter			
SystemTemplate::TransportProtocols::J1939TpConnection.maxBs			
Mapping Rule			**Mapping Type**
Please note that in the System Template the maximum block size is defined per J1939TpConnection. All J1939TpConnections in an ECU shall have the same value.			full

BSW Module	**BSW Context**	
J1939Tp	J1939Tp/J1939TpGeneral	
BSW Parameter		**BSW Type**
J1939TpRetrySupport		EcucBooleanParamDef
BSW Description		

This parameter is deprecated and will be removed in future. Old description: Enable support for repetition of failed transmission via TP.CM_CTS with a packet number that has already been sent. Retransmission is triggered when a sequence number is missing or a timeout occurs during reception. On sender side, retransmission is handled via the retry feature of PduR_J1939TpCopyTxData.		
M2 Template	**M2 Description**	
TPS_SYST	Enable support for protocol retry.	
M2 Parameter		
SystemTemplate::TransportProtocols::J1939TpConnection.retry		
Mapping Rule		**Mapping Type**
Please note that in the System Template the retry support is defined per J1939 TpConnection. All J1939TpConnections in an ECU shall have the same value.		full

BSW Module	**BSW Context**	
J1939Tp	J1939Tp/J1939TpGeneral	
BSW Parameter		**BSW Type**
J1939TpTxConfTimeout		EcucFloatParamDef
BSW Description		
Timeout in seconds for the CanIf Tx confirmation. After this time the J1939Tp assumes that an N-PDU could not be transmitted. Please note: The Tx confirmation timeout should be set to a value that enabled detection of a lost Tx confirmation in time, and that ensures that normal transmission delay caused by lower message priority does not lead to an error.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
J1939Tp	J1939Tp/J1939TpGeneral	
BSW Parameter		**BSW Type**
J1939TpVersionInfoApi		EcucBooleanParamDef
BSW Description		
The function J1939Tp_GetVersionInfo is configurable (On/Off) by this configuration parameter.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

D.3.2 J1939Nm Mapping

BSW Module	**BSW Context**	
J1939Nm	J1939Nm	
BSW Parameter		**BSW Type**

J1939NmConfigSet		EcucParamConfContainerDef
BSW Description		
This container is a MultipleConfigurationContainer, i.e. this container and its sub-containers exist once per configuration set.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
J1939Nm	J1939Nm/J1939NmConfigSet	
BSW Parameter		**BSW Type**
J1939NmChannel		EcucParamConfContainerDef
BSW Description		
Physical CAN channel handled by J1939Nm.		
M2 Template	**M2 Description**	
TPS_SYST	J1939 specific NmCluster attributes	
M2 Parameter		
SystemTemplate::NetworkManagement::J1939NmCluster		
Mapping Rule		**Mapping Type**
Container shall be created for each J1939NmCluster that is available in the Ecu Extract.		full

BSW Module	**BSW Context**	
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmChannel	
BSW Parameter		**BSW Type**
J1939NmComMNetworkHandleRef		EcucSymbolicNameReferenceDef
BSW Description		
Reference to the channel defined by the ComMChannel providing access to the unique channel index ComMChannelId.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmChannel	
BSW Parameter		**BSW Type**
J1939NmRxPdu		EcucParamConfContainerDef
BSW Description		
Contains the configuration of the I-PDU used to receive the AddressClaimed PG.		
M2 Template	**M2 Description**	
TPS_SYST		
M2 Parameter		
SystemTemplate::NetworkManagement::NmNode.rxNmPdu		
Mapping Rule		**Mapping Type**

Shall be derived from the NmPdu that is referenced by the NmNode.	full

BSW Module	BSW Context	
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmChannel/J1939NmRxPdu	
BSW Parameter		**BSW Type**
J1939NmRxPduId		EcucIntegerParamDef
BSW Description		
The I-PDU identifier used for RxIndication from CanIf.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmChannel/J1939NmRxPdu	
BSW Parameter		**BSW Type**
J1939NmRxPduRef		EcucReferenceDef
BSW Description		
Reference to the Pdu object representing the I-PDU.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmChannel	
BSW Parameter		**BSW Type**
J1939NmTxPdu		EcucParamConfContainerDef
BSW Description		
Contains the configuration of the I-PDU used to transmit the AddressClaimed PG.		
M2 Template	**M2 Description**	
TPS_SYST		
M2 Parameter		
SystemTemplate::NetworkManagement::NmNode.txNmPdu		
Mapping Rule		**Mapping Type**
Shall be derived from the NmPdu that is referenced by the NmNode.		full

BSW Module	BSW Context	
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmChannel/J1939NmTxPdu	
BSW Parameter		**BSW Type**
J1939NmTxPduId		EcucIntegerParamDef
BSW Description		
The I-PDU identifier used for TxConfirmation from CanIf.		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context	
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmChannel/J1939NmTxPdu	
BSW Parameter		**BSW Type**
J1939NmTxPduRef		EcucReferenceDef
BSW Description		
Reference to the Pdu object representing the I-PDU.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
J1939Nm	J1939Nm/J1939NmConfigSet	
BSW Parameter		**BSW Type**
J1939NmNode		EcucParamConfContainerDef
BSW Description		
Logical node representing one function handled by J1939Nm.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmNode	
BSW Parameter		**BSW Type**
J1939NmNodeChannelRef		EcucReferenceDef
BSW Description		
Reference to the channels this node has access to.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmNode	
BSW Parameter		**BSW Type**
J1939NmNodeId		EcucIntegerParamDef
BSW Description		

Unique identifier of this node. May be chosen to be identical with J1939NmNodePreferredAddress, if the same address shall not be used by different nodes on different channels.		
M2 Template	**M2 Description**	
TPS_SYST	Node identifier of local NmNode. Must be unique in the NmCluster.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmNode.nmNodeId		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmNode	
BSW Parameter		**BSW Type**
J1939NmNodeNameArbitraryAddressCapable		EcucBooleanParamDef
BSW Description		
Arbitrary Address Capable field of the NAME of this node.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmNode	
BSW Parameter		**BSW Type**
J1939NmNodeNameECUInstance		EcucIntegerParamDef
BSW Description		
ECU Instance field of the NAME of this node.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmNode	
BSW Parameter		**BSW Type**
J1939NmNodeNameFunction		EcucIntegerParamDef
BSW Description		
Function field of the NAME of this node.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**

J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmNode		
BSW Parameter		**BSW Type**	
J1939NmNodeNameFunctionInstance		EcucIntegerParamDef	
BSW Description			
Function Instance field of the NAME of this node.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmNode		
BSW Parameter		**BSW Type**	
J1939NmNodeNameIdentityNumber		EcucIntegerParamDef	
BSW Description			
Identity Number field of the NAME of this node.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmNode		
BSW Parameter		**BSW Type**	
J1939NmNodeNameIndustryGroup		EcucIntegerParamDef	
BSW Description			
Industry Group field of the NAME of this node.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmNode		
BSW Parameter		**BSW Type**	
J1939NmNodeNameManufacturerCode		EcucIntegerParamDef	
BSW Description			
Manufacturer Code field of the NAME of this node.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context	
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmNode	
BSW Parameter		**BSW Type**
J1939NmNodeNameVehicleSystem		EcucIntegerParamDef
BSW Description		
Vehicle System field of the NAME of this node.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmNode	
BSW Parameter		**BSW Type**
J1939NmNodeNameVehicleSystemInstance		EcucIntegerParamDef
BSW Description		
Vehicle System Instance field of the NAME of this node.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmNode	
BSW Parameter		**BSW Type**
J1939NmNodePreferredAddress		EcucIntegerParamDef
BSW Description		
Source address of this node used for address claiming.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
J1939Nm	J1939Nm/J1939NmConfigSet/J1939NmNode	
BSW Parameter		**BSW Type**
J1939NmNodeStartUpDelay		EcucBooleanParamDef
BSW Description		
If enabled, the communication will start after a delay of 250ms after transmission of the initial AddressClaimed. If disabled, communication will start immediately at start-up. Please note: According to J1939/81, the 250ms delay is not required for single address CAs with desired source addresses in the ranges 0..127 or 248..253.		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**	
J1939Nm	J1939Nm	
BSW Parameter		**BSW Type**
J1939NmDemEventParameterRefs		EcucParamConfContainerDef
BSW Description		
Container for the references to DemEventParameter elements which shall be invoked using the API Dem_ReportErrorStatus in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId value. The standardized errors are provided in the container and can be extended by vendor specific error references.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
J1939Nm	J1939Nm/J1939NmDemEventParameterRefs	
BSW Parameter		**BSW Type**
J1939NM_E_ADDRESS_LOST		EcucSymbolicNameReferenceDef
BSW Description		
Reference to the DemEventParameter which shall be issued when the ECU failed to claim one of its addresses.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
J1939Nm	J1939Nm	
BSW Parameter		**BSW Type**
J1939NmGeneral		EcucParamConfContainerDef
BSW Description		
Contains the general configuration parameters of the module.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context		
J1939Nm	J1939Nm/J1939NmGeneral		
BSW Parameter		**BSW Type**	
J1939NmBusOffDelayTickPeriod		EcucFloatParamDef	
BSW Description			
Duration of ticks that are used to time BusOff delays after conflicting address claims. This parameter should be synchronized with the main function period of the CAN State Manager.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
J1939Nm	J1939Nm/J1939NmGeneral		
BSW Parameter		**BSW Type**	
J1939NmDevErrorDetect		EcucBooleanParamDef	
BSW Description			
Pre-processor switch for enabling development error detection support.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
J1939Nm	J1939Nm/J1939NmGeneral		
BSW Parameter		**BSW Type**	
J1939NmMainFunctionPeriod		EcucFloatParamDef	
BSW Description			
Call cycle in seconds of J1939Nm_MainFunction.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context	
J1939Nm	J1939Nm/J1939NmGeneral	
BSW Parameter		**BSW Type**
J1939NmTxConfirmationTimeout		EcucFloatParamDef
BSW Description		
Time in seconds to wait for a confirmation after transmission of a message. The behaviour when the time elapses depends on the transmitted message.		
M2 Template	**M2 Description**	
M2 Parameter		

Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**	
J1939Nm	J1939Nm/J1939NmGeneral	
BSW Parameter		**BSW Type**
J1939NmUserCallout		EcucFunctionNameDef
BSW Description		
Pre-processor switch for enabling the <User_AddressClaimedIndication> and defining the name of the callout function.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
J1939Nm	J1939Nm/J1939NmGeneral	
BSW Parameter		**BSW Type**
J1939NmUserCalloutHeaderFile		EcucStringParamDef
BSW Description		
Header file which is included by J1939Nm when J1939NmUserCallout is enabled. This header file must provide the prototype of the <User_AddressClaimedIndication> defined in J1939NmUserCallout.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
J1939Nm	J1939Nm/J1939NmGeneral	
BSW Parameter		**BSW Type**
J1939NmVersionInfoApi		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling version info API support.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

D.4 FlexRay

D.4.1 FlexRay Driver Mapping

BSW Module	BSW Context	
Fr	Fr	
BSW Parameter		**BSW Type**
FrGeneral		EcucParamConfContainerDef
BSW Description		
General configuration (parameters) of the FlexRay Driver module.		
M2 Template	**M2 Description**	
TPS_SYST	FlexRay specific attributes to the physicalCluster	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster		
Mapping Rule		**Mapping Type**
Container must be created if the ECU is connected to a FlexRay Cluster		full

BSW Module	BSW Context	
Fr	Fr/FrGeneral	
BSW Parameter		**BSW Type**
FrBufferReconfig		EcucBooleanParamDef
BSW Description		
Please note that the parameter is deprecated and will be removed in future. Old description: Enables or disables buffer reconfiguration at runtime.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Fr	Fr/FrGeneral	
BSW Parameter		**BSW Type**
FrCtrlTestCount		EcucIntegerParamDef
BSW Description		
Maxmimum number of iterations the FlexRay controller hardware test is performed during controller initialization.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Fr	Fr/FrGeneral	
BSW Parameter		**BSW Type**

FrDevErrorDetect		EcucBooleanParamDef	
BSW Description			
Switches the Development Error Detection and Notification on or off. true: Development Error Detection and Notification enabled. false: Development Error Detection and Notification disabled.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
Fr	Fr/FrGeneral		
BSW Parameter		**BSW Type**	
FrDisableLPduSupport		EcucBooleanParamDef	
BSW Description			
Enables or disabled API function Fr_DisableLPdu.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
Fr	Fr/FrGeneral		
BSW Parameter		**BSW Type**	
FrIndex		EcucIntegerParamDef	
BSW Description			
Specifies the InstanceId of this module instance. If only one instance is present it shall have the Id 0.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**	
Fr	Fr/FrGeneral	
BSW Parameter		**BSW Type**
FrNumCtrlSupported		EcucIntegerParamDef
BSW Description		
Determines the maximum number of communication controllers that the driver supports.		
M2 Template	**M2 Description**	
M2 Parameter		

Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
Fr	Fr/FrGeneral
BSW Parameter	**BSW Type**
FrPrepareLPduSupport	EcucBooleanParamDef
BSW Description	
Enables or disables API function Fr_PrepareLPdu.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
Fr	Fr/FrGeneral
BSW Parameter	**BSW Type**
FrReconfigLPduSupport	EcucBooleanParamDef
BSW Description	
Enables or disabled API function Fr_ReconfigLPdu.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
Fr	Fr/FrGeneral
BSW Parameter	**BSW Type**
FrRxStringentCheck	EcucBooleanParamDef
BSW Description	
If stringent check is enabled (true), received frames are accepted only if no slot status error occurred.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
Fr	Fr/FrGeneral
BSW Parameter	**BSW Type**
FrRxStringentLengthCheck	EcucBooleanParamDef
BSW Description	
If stringent check is enabled (true), received frames are accepted only if the received payload length matches the configured payload length.	

M2 Template	M2 Description	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context		
Fr	Fr/FrGeneral		
BSW Parameter		**BSW Type**	
FrVersionInfoApi		EcucBooleanParamDef	
BSW Description			
Enables/disables the existence of the Fr_GetVersionInfo API.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
Fr	Fr		
BSW Parameter		**BSW Type**	
FrMultipleConfiguration		EcucParamConfContainerDef	
BSW Description			
Configuration of the individual controllers.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
Fr	Fr/FrMultipleConfiguration		
BSW Parameter		**BSW Type**	
FrController		EcucParamConfContainerDef	
BSW Description			
Configuration of the individual controller.			
M2 Template	**M2 Description**		
TPS_SYST	The communication controller is a dedicated hardware device by means of which hosts are sending frames to and receiving frames from the communication medium.		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController			
Mapping Rule			**Mapping Type**
Container must be created if the ECU contains a FlexRay communication controller that is connected to the regarded communication cluster.			full

BSW Module	BSW Context	
Fr	Fr/FrMultipleConfiguration/FrController	
BSW Parameter		**BSW Type**
FrAbsoluteTimer		EcucParamConfContainerDef
BSW Description		
Specifies the absolute timer configuration parameters of the Fr.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Fr	Fr/FrMultipleConfiguration/FrController/FrAbsoluteTimer	
BSW Parameter		**BSW Type**
FrAbsTimerIdx		EcucIntegerParamDef
BSW Description		
Contains the index of an absolute timer contained in Fr on a certain FlexRay CC.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Fr	Fr/FrMultipleConfiguration/FrController	
BSW Parameter		**BSW Type**
FrControllerDemEventParameterRefs		EcucParamConfContainerDef
BSW Description		
Container for the references to DemEventParameter elements which shall be invoked using the API Dem_ReportErrorStatus API in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId value. The standardized errors are provided in the container and can be extended by vendor specific error references.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Fr	Fr/FrMultipleConfiguration/FrController/FrControllerDemEventParameterRefs	
BSW Parameter		**BSW Type**
FR_E_CTRL_TESTRESULT		EcucSymbolicNameReferenceDef
BSW Description		
Reference to DEM event Id that is reported for FlexRay controller hardware test failure. If this parameter is not configured, no event reporting happens.		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrCtrlIdx	EcucIntegerParamDef
BSW Description	
Determines index of CC within Fr.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrFifo	EcucParamConfContainerDef
BSW Description	
One First In First Out (FIFO) queued receive structure, defining the admittance criteria to the FIFO, and mandating the ability to admit messages into the FIFO based on Message Id filtering criteria.	
M2 Template	**M2 Description**
TPS_SYST	One First In First Out (FIFO) queued receive structure, defining the admittance criteria to the FIFO, and mandating the ability to admit messages into the FIFO based on Message Id filtering criteria.
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayFifoConfiguration	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
Fr	Fr/FrMultipleConfiguration/FrController/FrFifo
BSW Parameter	**BSW Type**
FrAdmitWithoutMessageId	EcucBooleanParamDef
BSW Description	
Determines whether or not frames received in the dynamic segment that don't contain a message ID will be admitted into the FIFO.	
M2 Template	**M2 Description**
TPS_SYST	Boolean configuration which determines whether or not frames received in the dynamic segment that don't contain a message ID will be admitted into the FIFO.
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayFifoConfiguration.admitWithoutMessageId	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	BSW Context	
Fr	Fr/FrMultipleConfiguration/FrController/FrFifo	
BSW Parameter		**BSW Type**
FrBaseCycle		EcucIntegerParamDef
BSW Description		
FIFO cycle counter acceptance criteria.		
M2 Template	**M2 Description**	
TPS_SYST	FIFO cycle counter acceptance criteria.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayFifoConfiguration.baseCycle		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Fr	Fr/FrMultipleConfiguration/FrController/FrFifo	
BSW Parameter		**BSW Type**
FrChannels		EcucEnumerationParamDef
BSW Description		
FIFO channel admittance criteria.		
M2 Template	**M2 Description**	
TPS_SYST	The connection between the referencing ECU and the referenced channel via the referenced controller.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayFifoConfiguration.channel		
Mapping Rule		**Mapping Type**
If channelA is referenced set Parameter to FR_CHANNEL_A. If channelB is referenced set parameter to FR_CHANNEL_B. If two identical FlexrayFifoConfiguration elements exist with references to A and B only one FrFifo container shall be created (FR_CHANNEL_AB		full

BSW Module	BSW Context	
Fr	Fr/FrMultipleConfiguration/FrController/FrFifo	
BSW Parameter		**BSW Type**
FrCycleRepetition		EcucIntegerParamDef
BSW Description		
FIFO cycle counter acceptance criteria. Valid values are 1,2,4,5,8,10,16,20,32,40,50,64. Remark: Values 1,2,4,8,16,32,64 are valid only for FlexRay Protocol 2.1 Rev A compliance.		
M2 Template	**M2 Description**	
TPS_SYST	FIFO cycle counter acceptance criteria.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayFifoConfiguration.cycleRepetition		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Fr	Fr/FrMultipleConfiguration/FrController/FrFifo	
BSW Parameter		**BSW Type**
FrFifoDepth		EcucIntegerParamDef
BSW Description		
FrFifoDepth configures the maximum number of rx-frames which can be contained in the FIFO.		

M2 Template	M2 Description	
TPS_SYST	Fifo Depth.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayFifoConfiguration.fifoDepth		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Fr	Fr/FrMultipleConfiguration/FrController/FrFifo	
BSW Parameter		**BSW Type**
FrMsgIdMask		EcucIntegerParamDef
BSW Description		
FIFO message identifier acceptance criteria (Mask filter).		
M2 Template	**M2 Description**	
TPS_SYST	FIFO message identifier acceptance criteria (Mask filter).	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayFifoConfiguration.msgIdMask		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Fr	Fr/FrMultipleConfiguration/FrController/FrFifo	
BSW Parameter		**BSW Type**
FrMsgIdMatch		EcucIntegerParamDef
BSW Description		
FIFO message identifier acceptance criteria (Match filter).		
M2 Template	**M2 Description**	
TPS_SYST	FIFO message identifier acceptance criteria (Match filter).	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayFifoConfiguration.msgIdMatch		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Fr	Fr/FrMultipleConfiguration/FrController/FrFifo	
BSW Parameter		**BSW Type**
FrRange		EcucParamConfContainerDef
BSW Description		
FIFO Frame Id range acceptance criteria.		
M2 Template	**M2 Description**	
TPS_SYST	FIFO Frame Id range acceptance criteria.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayFifoRange		
Mapping Rule		**Mapping Type**
create container for each Fifo configuration		full

BSW Module	BSW Context	
Fr	Fr/FrMultipleConfiguration/FrController/FrFifo/FrRange	
BSW Parameter		**BSW Type**

FrRangeMax		EcucIntegerParamDef
BSW Description		
Last FrameId of this range that will be accepted by the FIFO.		
M2 Template	**M2 Description**	
TPS_SYST	Max Range.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayFifoRange.rangeMax		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Fr	Fr/FrMultipleConfiguration/FrController/FrFifo/FrRange	
BSW Parameter		**BSW Type**
FrRangeMin		EcucIntegerParamDef
BSW Description		
First FrameId of this range that will be accepted by the FIFO.		
M2 Template	**M2 Description**	
TPS_SYST	Min Range.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayFifoRange.rangeMin		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Fr	Fr/FrMultipleConfiguration/FrController	
BSW Parameter		**BSW Type**
FrPAllowHaltDueToClock		EcucBooleanParamDef
BSW Description		
Boolean flag that controls the transition to the POC:halt state due to a clock synchronization errors. If set to true, the CC is allowed to transition to POC:halt. If set to false, the CC will not transition to the POC:halt state but will enter or remain in the POC:normal passive state (self healing would still be possible)		
M2 Template	**M2 Description**	
TPS_SYST	Boolean flag that controls the transition to the POC:halt state due to a clock synchronization errors.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.allow HaltDueToClock		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Fr	Fr/FrMultipleConfiguration/FrController	
BSW Parameter		**BSW Type**
FrPAllowPassiveToActive		EcucIntegerParamDef
BSW Description		
Number of consecutive even/odd cycle pairs that must have valid clock correction terms before the CC will be allowed to transition from the POC:normal passive state to POC:normal active state. If set to zero, the CC is not allowed to transition from POC:normal passive to POC:normal active		
M2 Template	**M2 Description**	

TPS_SYST	Number of consecutive even/odd cycle pairs that must have valid clock correction terms before the Communication Controller will be allowed to transition from the POC:normal passive state to POC:normal active state.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.allowPassiveToActive		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Fr	Fr/FrMultipleConfiguration/FrController	
BSW Parameter		**BSW Type**
FrPChannels		EcucEnumerationParamDef
BSW Description		
Channels to which the node is connected. Implementation Type: Fr_ChannelType		
M2 Template	**M2 Description**	
TPS_SYST	Reference between the PhysicalChannel and ECUInstance via a CommunicationConnector.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreTopology::PhysicalChannel.commConnector		
Mapping Rule		**Mapping Type**
If channelA refers the connector set parameter to FR_CHANNEL_A. If channelB refers the connector set parameter to FR_CHANNEL_B. If channelA and channelB refer the connector set parameter to FR_CHANNEL_AB,		full

BSW Module	**BSW Context**	
Fr	Fr/FrMultipleConfiguration/FrController	
BSW Parameter		**BSW Type**
FrPClusterDriftDamping		EcucIntegerParamDef
BSW Description		
Local cluster drift damping factor used for rate correction [Microticks]. Remark: Upper limit 10 for FlexRay Protocol 3.0 compliance.		
M2 Template	**M2 Description**	
TPS_SYST	The cluster drift damping factor used in clock synchronization rate correction in microticks	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.clusterDriftDamping		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Fr	Fr/FrMultipleConfiguration/FrController	
BSW Parameter		**BSW Type**
FrPDecodingCorrection		EcucIntegerParamDef
BSW Description		
Value used by the receiver to calculate the difference between primary time reference point and secondary time reference point [Microticks]. Remark: Lower limit 14 for FlexRay Protocol 2.1 Rev. A compliance. Upper limit 136 for FlexRay Protocol 3.0 compliance.		

M2 Template	M2 Description	
TPS_SYST	Value used by the receiver to calculate the difference between primary time reference point and secondary time reference point. Unit: Microticks (pDecodingCorrection)	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.decodingCorrection		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context		
Fr	Fr/FrMultipleConfiguration/FrController		
BSW Parameter		**BSW Type**	
FrPDelayCompensationA		EcucIntegerParamDef	
BSW Description			
Value used to compensate for reception delays on the indicated channel. This covers assumed propagation delay up to cPropagationDelayMax for microticks in the range of 0.0125us to 0.05us [Microticks]. Remark: Lower limit 4 for FlexRay Protocol 3.0 compliance. Remark: Upper limit 200 for FlexRay Protocol 2.1 Rev A compliance.			
M2 Template	**M2 Description**		
TPS_SYST	Value used to compensate for reception delays on channel A Unit: Microticks		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.delay CompensationA			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	BSW Context		
Fr	Fr/FrMultipleConfiguration/FrController		
BSW Parameter		**BSW Type**	
FrPDelayCompensationB		EcucIntegerParamDef	
BSW Description			
Value used to compensate for reception delays on the indicated channel. This covers assumed propagation delay up to cPropagationDelayMax for microticks in the range of 0.0125us to 0.05us [Microticks]. Remark: Lower limit 4 for FlexRay Protocol 3.0 compliance. Remark: Upper limit 200 for FlexRay Protocol 2.1 Rev A compliance.			
M2 Template	**M2 Description**		
TPS_SYST	Value used to compensate for reception delays on channel B. Unit: Microticks		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.delay CompensationB			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	BSW Context	
Fr	Fr/FrMultipleConfiguration/FrController	
BSW Parameter		**BSW Type**
FrPExternalSync		EcucBooleanParamDef
BSW Description		

Flag indicating whether the node is externally synchronized (operating as time gateway sink in an TT-E cluster) or locally synchronized. If FrPExternalSync is set to 'true' then FrPTwoKeySlotMode must also be set to 'true'. Remarks: Set to 'false' for FlexRay Protocol 2.1 Rev. A compliance.	
M2 Template	**M2 Description**
TPS_SYST	Flag indicating whether the node is externally synchronized (operating as Time Gateway Sink in an TT-E Time Triggered External Sync cluster) or locally synchronized.
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.externalSync	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPFallBackInternal	EcucBooleanParamDef
BSW Description	
Flag indicating whether a time gateway sink node will switch to local clock operation when synchronization with the time gateway source node is lost (FrPFallBackInternal = true) or will instead go to POC:ready (FrPFallBackInternal =false). Remarks: Set to 'false' for FlexRay Protocol 2.1 Rev. A compliance.	
M2 Template	**M2 Description**
TPS_SYST	Flag indicating whether a Time Gateway Sink node will switch to local clock operation when synchronization with the Time Gateway Source node is lost (pFallBackInternal = true) or will instead go to POC:ready (pFallBackInternal = false).
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.fallBackInternal	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPKeySlotId	EcucIntegerParamDef
BSW Description	
ID of the key slot, i.e., the slot used to transmit the startup frame, sync frame, or designated key slot frame. If this parameter is set to zero the node does not have a key slot. For Fr3.0: if the value is not provided in System Description it shall be configured to 0. For Fr2.1: if the value is not provided in System Description it is driver implementation specific which value to configure.	
M2 Template	**M2 Description**
TPS_SYST	ID of the slot used to transmit the startup frame, sync frame, or designated single slot frame. For Fr3.0: if the value is not provided in SysT it shall be configured to 0. For Fr2.1: if the value is not provided in SysT it is driver impl. specific.
M2 Parameter	

SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.keySlotID	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**	
Fr	Fr/FrMultipleConfiguration/FrController	
BSW Parameter		**BSW Type**
FrPKeySlotOnlyEnabled		EcucBooleanParamDef
BSW Description		
Flag indicating whether or not the node shall enter key slot only mode following startup. Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter pSingleSlotEnabled.		
M2 Template	**M2 Description**	
TPS_SYST	Flag indicating whether or not the node shall enter single slot mode following startup.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.keySlotOnlyEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Fr	Fr/FrMultipleConfiguration/FrController	
BSW Parameter		**BSW Type**
FrPKeySlotUsedForStartup		EcucBooleanParamDef
BSW Description		
Flag indicating whether the key slot is used to transmit a startup frame. If FrPKeySlotUsedForStartup is set to true then FrPKeySlotUsedForSync must also be set to true. If FrPTwoKeySlotMode is set to true then both FrPKeySlotUsedForSync and FrPKeySlotUsedForStartup must also be set to true.		
M2 Template	**M2 Description**	
TPS_SYST	Flag indicating whether the Key Slot is used to transmit a startup frame.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.keySlotUsedForStartUp		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Fr	Fr/FrMultipleConfiguration/FrController	
BSW Parameter		**BSW Type**
FrPKeySlotUsedForSync		EcucBooleanParamDef
BSW Description		
Flag indicating whether the key slot is used to transmit a sync frame. If FrPKeySlotUsedForStartup is set to true then FrPKeySlotUsedForSync must also be set to true. If FrPTwoKeySlotMode is set to true then both FrPKeySlotUsedForSync and FrPKeySlotUsedForStartup must also be set to true.		
M2 Template	**M2 Description**	
TPS_SYST	Flag indicating whether the Key Slot is used to transmit a sync frame.	
M2 Parameter		

SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.keySlotUsedForSync	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**	
Fr	Fr/FrMultipleConfiguration/FrController	
BSW Parameter		**BSW Type**
FrPLatestTx		EcucIntegerParamDef
BSW Description		
Number of the last minislot in which a frame transmission can start in the dynamic segment. Remark: Upper limit 7980 for FlexRay Protocol 2.1 Rev A compliance.		
M2 Template	**M2 Description**	
TPS_SYST	The number of the last minislot in which a transmission can start in the dynamic segment for the respective node	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.latestTX		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Fr	Fr/FrMultipleConfiguration/FrController	
BSW Parameter		**BSW Type**
FrPMacroInitialOffsetA		EcucIntegerParamDef
BSW Description		
Integer number of macroticks between the static slot boundary and the following macrotick boundary of the secondary time reference point based on the nominal macrotick duration [Macroticks].		
M2 Template	**M2 Description**	
TPS_SYST	Integer number of macroticks between the static slot boundary and the closest macrotick boundary of the secondary time reference point based on the nominal macrotick duration. (pMacroInitialOffset)	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.macroInitialOffsetA		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Fr	Fr/FrMultipleConfiguration/FrController	
BSW Parameter		**BSW Type**
FrPMacroInitialOffsetB		EcucIntegerParamDef
BSW Description		
Integer number of macroticks between the static slot boundary and the following macrotick boundary of the secondary time reference point based on the nominal macrotick duration [Macroticks].		
M2 Template	**M2 Description**	
TPS_SYST	Integer number of macroticks between the static slot boundary and the following macrotick boundary of the secondary time reference point based on the nominal macrotick duration [Macroticks].	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.macroInitialOffsetB		

Mapping Rule	Mapping Type
1:1 mapping	full

BSW Module	BSW Context	
Fr	Fr/FrMultipleConfiguration/FrController	
BSW Parameter		**BSW Type**
FrPMicroInitialOffsetA		EcucIntegerParamDef
BSW Description		
Number of microticks between the secondary time reference point and the macrotick boundary immediately following the secondary time reference point. The parameter depends on FrPDelayCompensationA and therefore it has to be set independently for each channel [Microticks].		
M2 Template	**M2 Description**	
TPS_SYST	Number of microticks between the closest macrotick boundary described by gMacroInitialOffset and the secondary time reference point. The parameter depends on pDelayCompensationA and therefore it has to be set independently for each channel.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.microInitialOffsetA		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Fr	Fr/FrMultipleConfiguration/FrController	
BSW Parameter		**BSW Type**
FrPMicroInitialOffsetB		EcucIntegerParamDef
BSW Description		
Number of microticks between the secondary time reference point and the macrotick boundary immediately following the secondary time reference point. The parameter depends on FrPDelayCompensationB and therefore it has to be set independently for each channel [Microticks].		
M2 Template	**M2 Description**	
TPS_SYST	Number of microticks between the closest macrotick boundary described by gMacroInitialOffset and the secondary time reference point. The parameter depends on pDelayCompensationB and therefore it has to be set independently for each channel.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.microInitialOffsetB		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Fr	Fr/FrMultipleConfiguration/FrController	
BSW Parameter		**BSW Type**
FrPMicroPerCycle		EcucIntegerParamDef
BSW Description		

Nominal number of microticks in the communication cycle of the local node. If nodes have different microtick durations this number will differ from node to node [Microticks]. Remark: Lower limit 960 for FlexRay Protocol 3.0 compliance. Upper limit 640000 for FlexRay Protocol 2.1 Rev A compliance.	
M2 Template	**M2 Description**
TPS_SYST	The nominal number of microticks in a communication cycle
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.microPerCycle	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPNmVectorEarlyUpdate	EcucBooleanParamDef
BSW Description	
Flag indicating when the update of the Network Management Vector in the CHI shall take place. If FrPNmVectorEarlyUpdate is set to false, the update shall take place after the NIT. If FrPNmVectorEarlyUpdate is set to true, the update shall take place after the end of the static segment. Remarks: Set to 'false' for FlexRay Protocol 2.1 Rev. A compliance.	
M2 Template	**M2 Description**
TPS_SYST	Flag indicating when the update of the Network Management Vector in the CHI shall take place.
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.nmVectorEarlyUpdate	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPOffsetCorrectionOut	EcucIntegerParamDef
BSW Description	
Magnitude of the maximum permissible offset correction value [Microticks]. Remark: Upper limit 15567 for FlexRay Protocol 2.1 Rev A compliance. Remark: Lower limit 15 for FlexRay Protocol 3.0 compliance.	
M2 Template	**M2 Description**
TPS_SYST	Magnitude of the maximum permissible offset correction value. Unit:microtick (pOffsetCorrectionOut)
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.offsetCorrectionOut	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**

FrPOffsetCorrectionStart		EcucIntegerParamDef	
BSW Description			
Start of the offset correction phase within the NIT, expressed as the number of macroticks from the start of cycle [Macroticks]. Remark: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gOffsetCorrectionStart. Remark: Lower limit 9 for FlexRay Protocol 2.1 Rev A compliance.			
M2 Template	**M2 Description**		
TPS_SYST	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		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.offsetCorrectionStart			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	**BSW Context**		
Fr	Fr/FrMultipleConfiguration/FrController		
BSW Parameter		**BSW Type**	
FrPPayloadLengthDynMax		EcucIntegerParamDef	
BSW Description			
Maximum payload length for dynamic frames [16 bit words].			
M2 Template	**M2 Description**		
TPS_SYST	Maximum payload length for the dynamic channel of a frame in 16 bit WORDS.		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.maximumDynamicPayloadLength			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	**BSW Context**		
Fr	Fr/FrMultipleConfiguration/FrController		
BSW Parameter		**BSW Type**	
FrPRateCorrectionOut		EcucIntegerParamDef	
BSW Description			
Magnitude of the maximum permissible rate correction value and the maximum drift offset between two nodes operating with unsynchronized clocks for one communication cycle [Microticks]. Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter pdMaxDrift. Lower limit 3 for FlexRay Protocol 3.0 compliance. Upper limit 1923 for FlexRay Protocol 2.1 Rev A compliance.			
M2 Template	**M2 Description**		
TPS_SYST	Magnitude of the maximum permissible rate correction value and the maximum drift offset between two nodes operating with unsynchronized clocks for one communication cycle. Unit:Microticks (pRateCorrectionOut)		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.rateCorrectionOut			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	**BSW Context**	
Fr	Fr/FrMultipleConfiguration/FrController	
BSW Parameter		**BSW Type**

FrPSamplesPerMicrotick	EcucEnumerationParamDef
BSW Description	
Number of samples per microtick. Remark: Allowed range N1SAMPLES, N2SAMPLES for FlexRay Protocol 3.0 compliance.	
M2 Template	**M2 Description**
TPS_SYST	Number of samples per microtick
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.samplesPerMicrotick	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPSecondKeySlotId	EcucIntegerParamDef
BSW Description	
ID of the second key slot, in which a second startup frame shall be sent when operating as a coldstart node in a TT-L or TT-D cluster. If this parameter is set to zero the node does not have a second key slot. Remark: Set to 0 for FlexRay Protocol 2.1 Rev A compliance.	
M2 Template	**M2 Description**
TPS_SYST	ID of the second Key slot, in which a second startup frame shall be sent in TT-L Time Triggered Local Master Sync or TT-E Time Triggered External Sync mode. If this parameter is set to zero the node does not have a second key slot.
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.secondKeySlotId	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
Fr	Fr/FrMultipleConfiguration/FrController
BSW Parameter	**BSW Type**
FrPTwoKeySlotMode	EcucBooleanParamDef
BSW Description	
Flag indicating whether node operates as a coldstart node in a TT-E or TT-L cluster. If pTwoKeySlotMode is set to true then both pKeySlotUsedForSync and pKeySlotUsedForStartup must also be set to true. If pExternalSync is set to true then pTwoKeySlotMode must also be set to true. Remark: Set to false for FlexRay Protocol 2.1 Rev A compliance.	
M2 Template	**M2 Description**
TPS_SYST	Flag indicating whether node operates as a startup node in a TT-E Time Triggered External Sync or TT-L Time Triggered Local Master Sync cluster.
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.twoKeySlotMode	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**

Fr	Fr/FrMultipleConfiguration/FrController	
BSW Parameter		**BSW Type**
FrPWakeupChannel		EcucEnumerationParamDef
BSW Description		
Channel used by the node to send a wakeup pattern. FrPWakeupChannel must be selected from among the channels configured by FrPChannels.		
M2 Template	**M2 Description**	
TPS_SYST	Referenced channel used by the node to send a wakeup pattern. (pWakeupChannel)	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationConnector.wakeUpChannel		
Mapping Rule		**Mapping Type**
If channelA refers to the FlexrayCommunicationConnector and wakeUpChannel=true then FrPWakeupChannel = FR_CHANNEL_A. If channelB refers to the FlexrayCommunicationConnector and wakeUpChannel = true then FrPWakeup Chanel = FR_CHANNEL_B.		full

BSW Module	**BSW Context**	
Fr	Fr/FrMultipleConfiguration/FrController	
BSW Parameter		**BSW Type**
FrPWakeupPattern		EcucIntegerParamDef
BSW Description		
Number of repetitions of the wakeup symbol that are combined to form a wakeup pattern when the node enters the POC:wakeup send state. Remark: Lower limit 2 for FlexRay Protocol 2.1 Rev A compliance.		
M2 Template	**M2 Description**	
TPS_SYST	Number of repetitions of the Tx-wakeup symbol to be sent during the CC_WakeupSend state of this Node in the cluster	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.wakeUpPattern		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Fr	Fr/FrMultipleConfiguration/FrController	
BSW Parameter		**BSW Type**
FrPdAcceptedStartupRange		EcucIntegerParamDef
BSW Description		
Expanded range of measured clock deviation allowed for startup frames during integration [Microticks]. Remark: Upper limit 1875 for FlexRay Protocol 2.1 Rev A compliance. Remark: Lower limit 29 for FlexRay Protocol 3.0 compliance.		
M2 Template	**M2 Description**	
TPS_SYST	Expanded range of measured clock deviation allowed for startup frames during integration. Unit:microtick	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.acceptedStartupRange		
Mapping Rule		**Mapping Type**
1:1 mapping		full

<table>
<tr><th>BSW Module</th><th colspan="2">BSW Context</th></tr>
<tr><td>Fr</td><td colspan="2">Fr/FrMultipleConfiguration/FrController</td></tr>
<tr><th colspan="2">BSW Parameter</th><th>BSW Type</th></tr>
<tr><td colspan="2">FrPdListenTimeout</td><td>EcucIntegerParamDef</td></tr>
<tr><th colspan="3">BSW Description</th></tr>
<tr><td colspan="3">Value for the startup listen timeout and wakeup listen timeout. Although this is a node local parameter, the real time equivalent of this value should be the same for all nodes in the cluster [Microticks].
Remark: Lower limit 1926 for FlexRay Protocol 3.0 compliance.
Upper limit 1283846 for FlexRay Protocol 2.1 Rev. A compliance.</td></tr>
<tr><th>M2 Template</th><th colspan="2">M2 Description</th></tr>
<tr><td>TPS_SYST</td><td colspan="2">Value for the startup listen timeout and wakeup listen timeout. Although this is a node local parameter, the real time equivalent of this value should be the same for all nodes in the cluster. Unit: Microticks</td></tr>
<tr><th colspan="3">M2 Parameter</th></tr>
<tr><td colspan="3">SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.listenTimeout</td></tr>
<tr><th colspan="2">Mapping Rule</th><th>Mapping Type</th></tr>
<tr><td colspan="2">1:1 mapping</td><td>full</td></tr>
</table>

<table>
<tr><th>BSW Module</th><th colspan="2">BSW Context</th></tr>
<tr><td>Fr</td><td colspan="2">Fr/FrMultipleConfiguration/FrController</td></tr>
<tr><th colspan="2">BSW Parameter</th><th>BSW Type</th></tr>
<tr><td colspan="2">FrPdMicrotick</td><td>EcucEnumerationParamDef</td></tr>
<tr><th colspan="3">BSW Description</th></tr>
<tr><td colspan="3">Duration of a microtick.
Remark: Allowed range T12_5NS, T25NS, T50NS for FlexRay Protocol 3.0 compliance.</td></tr>
<tr><th>M2 Template</th><th colspan="2">M2 Description</th></tr>
<tr><td>TPS_SYST</td><td colspan="2">Duration of a microtick. This attribute can be derived from samplePerMicrotick and gdSampleClockPeriod. Unit: seconds</td></tr>
<tr><th colspan="3">M2 Parameter</th></tr>
<tr><td colspan="3">SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController.microtickDuration</td></tr>
<tr><th colspan="2">Mapping Rule</th><th>Mapping Type</th></tr>
<tr><td colspan="2">1:1 mapping</td><td>full</td></tr>
</table>

D.4.2 FlexRay Interface Mapping

<table>
<tr><th>BSW Module</th><th colspan="2">BSW Context</th></tr>
<tr><td>FrIf</td><td colspan="2">FrIf</td></tr>
<tr><th colspan="2">BSW Parameter</th><th>BSW Type</th></tr>
<tr><td colspan="2">FrIfConfig</td><td>EcucParamConfContainerDef</td></tr>
<tr><th colspan="3">BSW Description</th></tr>
<tr><td colspan="3">Configuration of the FlexRay Interface. This container is a MultipleConfigurationContainer, i.e. this container and its sub-containers exist once per configuration set.</td></tr>
<tr><th>M2 Template</th><th colspan="2">M2 Description</th></tr>
<tr><td>TPS_SYST</td><td colspan="2">The CommunicationCluster is the main element to describe the topological connection</td></tr>
<tr><th colspan="3">M2 Parameter</th></tr>
<tr><td colspan="3">SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster</td></tr>
<tr><th colspan="2">Mapping Rule</th><th>Mapping Type</th></tr>
<tr><td colspan="2">Container must be created if the ECU is connected to a FlexRay Cluster</td><td>full</td></tr>
</table>

BSW Module	BSW Context		
FrIf	FrIf/FrIfConfig		
BSW Parameter		**BSW Type**	
FrIfCluster		EcucParamConfContainerDef	
BSW Description			
This container specifies a FrIf Cluster and all related data which is required to enable communication of the Cluster. A Cluster may consist of more than one Controller.			
M2 Template	**M2 Description**		
TPS_SYST	The CommunicationCluster is the main element to describe the topological connection		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster			
Mapping Rule			**Mapping Type**
Container must be created if the ECU is connected to a FlexRay Cluster			full

BSW Module	BSW Context		
FrIf	FrIf/FrIfConfig/FrIfCluster		
BSW Parameter		**BSW Type**	
FrIfClstIdx		EcucIntegerParamDef	
BSW Description			
This parameter provides a zero-based consecutive index of the FlexRay Clusters. Upper layer BSW modules and the FrIf itself use this index to identify a FlexRay Cluster.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
FrIf	FrIf/FrIfConfig/FrIfCluster		
BSW Parameter		**BSW Type**	
FrIfClusterDemEventParameterRefs		EcucParamConfContainerDef	
BSW Description			
Container for the references to DemEventParameter elements which shall be invoked using the API Dem_ReportErrorStatus API in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId value. The standardized errors are provided in the container and can be extended by vendor specific error references.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfClusterDemEventParameterRefs	
BSW Parameter		**BSW Type**
FRIF_E_ACS_CH_A		EcucSymbolicNameReferenceDef
BSW Description		

Reference to the DemEventParameter which shall be issued when an error in ACS on channel A was detected. If the reference is not configured the error shall not be reported (neither to DET nor to DEM).	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfClusterDemEventParameterRefs
BSW Parameter	**BSW Type**
FRIF_E_ACS_CH_B	EcucSymbolicNameReferenceDef
BSW Description	
Reference to the DemEventParameter which shall be issued when an error in ACS on channel B was detected. If the reference is not configured the error shall not be reported (neither to DET nor to DEM).	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfClusterDemEventParameterRefs
BSW Parameter	**BSW Type**
FRIF_E_NIT_CH_A	EcucSymbolicNameReferenceDef
BSW Description	
Reference to the DemEventParameter which shall be issued when an error in NIT on channel A was detected. If the reference is not configured the error shall not be reported (neither to DET nor to DEM).	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfClusterDemEventParameterRefs
BSW Parameter	**BSW Type**
FRIF_E_NIT_CH_B	EcucSymbolicNameReferenceDef
BSW Description	
Reference to the DemEventParameter which shall be issued when an error in NIT on channel B was detected. If the reference is not configured the error shall not be reported (neither to DET nor to DEM).	
M2 Template	**M2 Description**

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfClusterDemEventParameterRefs	
BSW Parameter		**BSW Type**
FRIF_E_SW_CH_A		EcucSymbolicNameReferenceDef
BSW Description		
Reference to the DemEventParameter which shall be issued when an error in SW on channel A was detected. If the reference is not configured the error shall not be reported (neither to DET nor to DEM).		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfClusterDemEventParameterRefs	
BSW Parameter		**BSW Type**
FRIF_E_SW_CH_B		EcucSymbolicNameReferenceDef
BSW Description		
Reference to the DemEventParameter which shall be issued when an error in SW on channel B was detected. If the reference is not configured the error shall not be reported (neither to DET nor to DEM).		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfController		EcucParamConfContainerDef
BSW Description		
This container contains the configuration of FlexRay CC.		
M2 Template	**M2 Description**	
TPS_SYST	The communication controller is a dedicated hardware device by means of which hosts are sending frames to and receiving frames from the communication medium.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationController		
Mapping Rule		**Mapping Type**
Container must be created if the ECU contains a FlexRay communication controller that is connected to the regarded communication cluster.		full

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfController	
BSW Parameter		**BSW Type**
FrIfCtrlIdx		EcucIntegerParamDef
BSW Description		
This parameter provides a zero-based consecutive index of the FlexRay Communication Controllers. Upper layer BSW modules and the FrIf itself use this index to identify a FlexRay CC.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfController	
BSW Parameter		**BSW Type**
FrIfFrCtrlRef		EcucSymbolicNameReferenceDef
BSW Description		
Reference to a Controller, which is handled by a specific Driver. This reference is unique for the ECU.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfController	
BSW Parameter		**BSW Type**
FrIfFrameTriggering		EcucParamConfContainerDef
BSW Description		
A Frame triggering contains the communication parameters of the FlexRay Frame as well as a reference to the Frame Construction Plan.		
M2 Template	**M2 Description**	
TPS_SYST	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.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayCommunication::FlexrayFrameTriggering		
Mapping Rule		**Mapping Type**
Container must be created for each existing FlexRayFrameTriggering element that is connected to a CommConnectorPort of the regarded communication controller.		full

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfController/FrIfFrameTriggering	
BSW Parameter		**BSW Type**
FrIfAllowDynamicLSduLength		EcucBooleanParamDef

BSW Description	
Allows L-PDU length reduction ('FrIfLSduLength' defines max. length) and indicates that the related CC buffer has to be reconfigured for the actual length and Header-CRC before transmission of the L-PDU.	
M2 Template	**M2 Description**
TPS_SYST	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.
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayCommunication::FlexrayFrameTriggering.allowDynamicLSduLength	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	BSW Context
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfController/FrIfFrameTriggering
BSW Parameter	**BSW Type**
FrIfAlwaysTransmit	EcucBooleanParamDef
BSW Description	
Defines whether the driver's API function Fr_TransmitTxLPdu() shall always be called for this L-PDU.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfController/FrIfFrameTriggering
BSW Parameter	**BSW Type**
FrIfBaseCycle	EcucIntegerParamDef
BSW Description	
This parameter contains the FlexRay Base Cycle used to transmit this FlexRay Frame.	
M2 Template	**M2 Description**
TPS_SYST	The first communication cycle where the frame is sent. This value is incremented at the beginning of each new cycle, ranging from 0 to 63, and is reset to 0 after a sequence of 64 cycles.
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreTopology::CycleRepetition.BaseCycle	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	BSW Context
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfController/FrIfFrameTriggering
BSW Parameter	**BSW Type**
FrIfChannel	EcucEnumerationParamDef
BSW Description	
This parameter contains the FlexRay Channel used to transmit this FlexRay Frame.	
M2 Template	**M2 Description**
TPS_SYST	One frame triggering is defined for exactly one channel. Channels may have assigned an arbitrary number of frame triggerings.

M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreTopology::PhysicalChannel.frameTriggering	
Mapping Rule	**Mapping Type**
FrameTriggering element in the System Template is aggregated by the Physical Channel that is used to transmit this FlexRay Frame	full

BSW Module	BSW Context		
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfController/FrIfFrameTriggering		
BSW Parameter		**BSW Type**	
FrIfCycleRepetition		EcucIntegerParamDef	
BSW Description			
This parameter contains the FlexRay Cycle Repetition used to transmit this FlexRay Frame.. possible Values: 1,2,4,8,16,32,64			
M2 Template	**M2 Description**		
TPS_SYST	The number of communication cycles (after the first cycle) whenever the frame described by this timing is sent again.		
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreTopology::CycleRepetition.CycleRepetition			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	BSW Context		
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfController/FrIfFrameTriggering		
BSW Parameter		**BSW Type**	
FrIfFrameStructureRef		EcucReferenceDef	
BSW Description			
Reference to the Construction Plan of the FlexRay Frame.			
M2 Template	**M2 Description**		
TPS_SYST	One frame can be triggered 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.		
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreCommunication::FrameTriggering.frame			
Mapping Rule			**Mapping Type**
Reference must comply to the reference in the System Description between the FrameTriggering element and the Frame.element			full

BSW Module	BSW Context		
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfController/FrIfFrameTriggering		
BSW Parameter		**BSW Type**	
FrIfFrameTriggeringDemEventParameterRefs		EcucParamConfContainerDef	
BSW Description			
Container for the references to DemEventParameter elements which shall be invoked using the API Dem_ReportErrorStatus API in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId value. The standardized errors are provided in the container and can be extended by vendor specific error references.			
M2 Template	**M2 Description**		
M2 Parameter			

Mapping Rule	Mapping Type
	local

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfController/FrIfFrameTriggering/FrIfFrameTriggeringDemEventParameterRefs	
BSW Parameter		**BSW Type**
FRIF_E_LPDU_SLOTSTATUS		EcucSymbolicNameReferenceDef
BSW Description		
Reference to DEM event Id that is reported when FlexRay driver module detects slot errors. If this parameter is not configured, no event reporting happens.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfController/FrIfFrameTriggering	
BSW Parameter		**BSW Type**
FrIfLSduLength		EcucIntegerParamDef
BSW Description		
The payload length of the Frame is given here. This parameter is required for validation if configured PDUs and update information fits into the Frame at configuration time [bytes].		
M2 Template	**M2 Description**	
TPS_SYST	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).	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::Frame.frameLength		
Mapping Rule		**Mapping Type**
Find Frame that is referenced by the regarded FrameTriggering and use the frameLength attribute		full

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfController/FrIfFrameTriggering	
BSW Parameter		**BSW Type**
FrIfMessageId		EcucIntegerParamDef
BSW Description		
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.		
M2 Template	**M2 Description**	
TPS_SYST	The first two bytes of the payload segment of the FlexRay frame format for frames transmitted in the	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayCommunication::FlexrayFrameTriggering.messageId		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfController/FrIfFrameTriggering	
BSW Parameter		**BSW Type**
FrIfPayloadPreamble		EcucBooleanParamDef
BSW Description		
Switching the Payload Preamble bit.		
M2 Template	**M2 Description**	
TPS_SYST	Switching the Payload Preamble bit.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayCommunication::FlexrayFrameTriggering.payloadPreambleIndicator		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfController/FrIfFrameTriggering	
BSW Parameter		**BSW Type**
FrIfSlotId		EcucIntegerParamDef
BSW Description		
This parameter contains the FlexRay Slot ID used to transmit this FlexRay Frame.		
M2 Template	**M2 Description**	
TPS_SYST	In the static part the SlotID defines the slot in which the frame is transmitted. In the dynamic part, the slot id is equivalent to a priority.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayCommunication::FlexrayAbsolutelyScheduledTiming.slotID		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfController	
BSW Parameter		**BSW Type**
FrIfLPdu		EcucParamConfContainerDef
BSW Description		
Reference to a L-PDU index		
M2 Template	**M2 Description**	
TPS_SYST	Data frame which is sent over a communication medium. This element describes the pure Layout of a frame sent on a channel.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::Frame		
Mapping Rule		**Mapping Type**
Create container for each FlexRay Frame that is transmitted or received via the regarded communication controller..		full

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfController/FrIfLPdu	
BSW Parameter		**BSW Type**
FrIfLPduIdx		EcucIntegerParamDef
BSW Description		
This parameter identifies the L-PDU in the interaction between FlexRay Interface and FlexRay Driver.		

M2 Template	M2 Description	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context		
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfController/FrIfLPdu		
BSW Parameter		**BSW Type**	
FrIfReconfigurable		EcucBooleanParamDef	
BSW Description			
This parameter specifies that this LPdu is reconfigurable using FrIf_ReconfigLPdu. This means that this LPdu can be assigned to a different FrameTriggering at runtime. However, this reconfiguration is limited by hardware constraints. The direction of the LPdu cannot be reconfigured.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfController/FrIfLPdu		
BSW Parameter		**BSW Type**	
FrIfVBTriggeringRef		EcucReferenceDef	
BSW Description			
Reference to the assigned Frame triggering.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfController		
BSW Parameter		**BSW Type**	
FrIfTransceiver		EcucParamConfContainerDef	
BSW Description			
Up to two FlexRay Transceivers may connect a Controller to a Cluster. This container realizes a Controller-Transceiver assignment.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfController/FrIfTransceiver	
BSW Parameter		**BSW Type**
FrIfClusterChannel		EcucEnumerationParamDef
BSW Description		
This parameter identifies to which one of the two Channels (A, B, A and B) of the Cluster the Transceiver is connected. FrIfClusterChannel shall map to Fr_ChannelType: FRIF_CHANNEL_A == FR_CHANNEL_A FRIF_CHANNEL_B == FR_CHANNEL_B FR_CHANNEL_AB shall not be used.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfController/FrIfTransceiver	
BSW Parameter		**BSW Type**
FrIfFrTrcvChannelRef		EcucSymbolicNameReferenceDef
BSW Description		
Reference to a Transceiver Driver Channel. This reference is unique for the ECU.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfDetectNITError		EcucBooleanParamDef
BSW Description		
Indicates whether NIT error status of each cluster shall be detected or not.		
M2 Template	**M2 Description**	
TPS_SYST	Indicates whether NIT error status of each cluster shall be detected or not.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.detectNitError		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGChannels		EcucEnumerationParamDef
BSW Description		
The channels that are used by the cluster. Implementation Type: Fr_ChannelType		
M2 Template	**M2 Description**	

TPS_SYST	A physical channel is the transmission medium that is used to send and receive information between two communicating ECUs.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayPhysicalChannel		
Mapping Rule		**Mapping Type**
The channels that are used by the cluster are described in the System Template by the CommunicationCluster-PhysicalChannel relationship.		full

BSW Module	**BSW Context**	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGColdStartAttempts		EcucIntegerParamDef
BSW Description		
Maximum number of times a node in the cluster is permitted to attempt to start the cluster by initiating schedule synchronization		
M2 Template	**M2 Description**	
TPS_SYST	The maximum number of times that a node in this cluster is permitted to attempt to start the cluster by initiating schedule synchronization	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.coldStartAttempts		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGCycleCountMax		EcucIntegerParamDef
BSW Description		
Maximum cycle counter value in a given cluster. Remark: Set to 63 for FlexRay Protocol 2.1 Rev. A compliance.		
M2 Template	**M2 Description**	
TPS_SYST	Maximum cycle counter value in a given cluster.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.cycleCountMax		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGListenNoise		EcucIntegerParamDef
BSW Description		
Upper limit for the start up listen timeout and wake up listen timeout in the presence of noise. It is used as a multiplier of the node parameter pdListenTimeout.		
M2 Template	**M2 Description**	
TPS_SYST	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	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.listenNoise		
Mapping Rule		**Mapping Type**

1:1 mapping	full

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGMacroPerCycle		EcucIntegerParamDef
BSW Description		
Number of macroticks in a communication cycle. Note: Lower limit 10 for FlexRay Protocol 2.1 Rev. A compliance		
M2 Template	**M2 Description**	
TPS_SYST	The number of macroticks in a communication cycle	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.macroPerCycle		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGMaxWithoutClockCorrectFatal		EcucIntegerParamDef
BSW Description		
Threshold used for testing the vClockCorrectionFailed counter. 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. [Even/odd cycle pairs].		
M2 Template	**M2 Description**	
TPS_SYST	Threshold used for testing the vClockCorrectionFailed counter.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.maxWithoutClockCorrectionFatal		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGMaxWithoutClockCorrectPassive		EcucIntegerParamDef
BSW Description		
Threshold used for testing the vClockCorrectionFailed counter. 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. [Even/Odd cycle pairs]		
M2 Template	**M2 Description**	
TPS_SYST	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.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.maxWithoutClockCorrectionPassive		
Mapping Rule		**Mapping Type**

1:1 mapping	full

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGNetworkManagementVectorLength		EcucIntegerParamDef
BSW Description		
Length of the Network Management vector in a cluster [bytes]		
M2 Template	**M2 Description**	
TPS_SYST	Length of the Network Management vector on a cluster. Unit: Bytes	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.networkManagementVectorLength		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGNumberOfMinislots		EcucIntegerParamDef
BSW Description		
Number of minislots in the dynamic segment Remark: Upper limit 7986 for FlexRay Protocol 2.1 Rev. A compliance		
M2 Template	**M2 Description**	
TPS_SYST	Number of Minislots in the dynamic segment.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.numberOfMinislots		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGNumberOfStaticSlots		EcucIntegerParamDef
BSW Description		
Number of static slots in the static segment		
M2 Template	**M2 Description**	
TPS_SYST	The number of static slots in the static segment.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.numberOfStaticSlots		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGPayloadLengthStatic		EcucIntegerParamDef
BSW Description		

Payload length of a static frame [16 bit words]		
M2 Template	**M2 Description**	
TPS_SYST	Globally configured payload length of a static frame. Unit: 16-bit WORDS.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.payloadLengthStatic		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**		
FrIf	FrIf/FrIfConfig/FrIfCluster		
BSW Parameter		**BSW Type**	
FrIfGSyncFrameIDCountMax		EcucIntegerParamDef	
BSW Description			
Maximum number of distinct syncframe identifiers present in a given cluster. This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gSyncNodeMax.			
M2 Template	**M2 Description**		
TPS_SYST	Maximum number of distinct syncframe identifiers present in a given cluster.		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.syncFrameIdCountMax			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	**BSW Context**		
FrIf	FrIf/FrIfConfig/FrIfCluster		
BSW Parameter		**BSW Type**	
FrIfGdActionPointOffset		EcucIntegerParamDef	
BSW Description			
Number of macroticks the action point is offset from the beginning of a static slot.			
M2 Template	**M2 Description**		
TPS_SYST	The offset of the action point in networks		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.actionPointOffset			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	**BSW Context**		
FrIf	FrIf/FrIfConfig/FrIfCluster		
BSW Parameter		**BSW Type**	
FrIfGdBit		EcucEnumerationParamDef	
BSW Description			
Nominal bit time in seconds			
M2 Template	**M2 Description**		
TPS_SYST	Nominal bit time (= 1 / fx:SPEED). gdBit = cSamplesPerBit * gdSampleClock-Period. Unit: seconds (gdBit)		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.bit			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGdCasRxLowMax		EcucIntegerParamDef
BSW Description		
Upper limit of the CAS acceptance windows [gdBit] Remark: Range 67 to 99 for FlexRay Protocol 2.1 Rev. A compliance		
M2 Template	**M2 Description**	
TPS_SYST	Upper limit of the Collision Avoidance Symbol (CAS) acceptance window. Unit:bitDuration	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.casRxLowMax		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGdCycle		EcucFloatParamDef
BSW Description		
Length of the cycle, expressed in [s] Remark: Lower limit 0.000024 for FlexRay Protocol 3.0 compliance.		
M2 Template	**M2 Description**	
TPS_SYST	Length of the cycle. Unit: seconds	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.cycle		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGdDynamicSlotIdlePhase		EcucIntegerParamDef
BSW Description		
Duration of the idle phase within a dynamic slot [Minislots].		
M2 Template	**M2 Description**	
TPS_SYST	The duration of the dynamic slot idle phase in minislots.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.dynamicSlotIdlePhase		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGdIgnoreAfterTx		EcucIntegerParamDef
BSW Description		
Duration for which the bitstrobing is paused after transmission [gdBit]. Remark: Set to 0 for FlexRay Protocol 2.1 Rev. A compliance.		

M2 Template	M2 Description	
TPS_SYST	Duration for which the bitstrobing is paused after transmission [gdBit].	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.ignoreAfterTx		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context		
FrIf	FrIf/FrIfConfig/FrIfCluster		
BSW Parameter		**BSW Type**	
FrIfGdMacrotick		EcucFloatParamDef	
BSW Description			
Duration of the cluster wide nominal macrotick, expressed in s			
M2 Template	**M2 Description**		
TPS_SYST	Duration of the cluster wide nominal macrotick, expressed in seconds		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.macrotickDuration			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	BSW Context		
FrIf	FrIf/FrIfConfig/FrIfCluster		
BSW Parameter		**BSW Type**	
FrIfGdMiniSlotActionPointOffset		EcucIntegerParamDef	
BSW Description			
Number of Macroticks the Minislot action point is offset from the beginning of a Minislot [Macroticks].			
M2 Template	**M2 Description**		
TPS_SYST	The Offset of the action point within a minislot. Unit: macroticks		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.minislotActionPointOffset			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	BSW Context		
FrIf	FrIf/FrIfConfig/FrIfCluster		
BSW Parameter		**BSW Type**	
FrIfGdMinislot		EcucIntegerParamDef	
BSW Description			
Duration of a minislot [Macroticks]			
M2 Template	**M2 Description**		
TPS_SYST	The duration of a minislot (dynamic segment). Unit: macroticks.		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.minislotDuration			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**

FrIfGdNit	EcucIntegerParamDef
BSW Description	
Duration of the Network Idle Time [Macroticks] Remark: Upper limit 805 for FlexRay Protocol 2.1 Rev. A compliance.	
M2 Template	**M2 Description**
TPS_SYST	The duration of the network idle time in macroticks
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.networkIdleTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
FrIf	FrIf/FrIfConfig/FrIfCluster
BSW Parameter	**BSW Type**
FrIfGdSampleClockPeriod	EcucEnumerationParamDef
BSW Description	
Sample clock period	
M2 Template	**M2 Description**
TPS_SYST	Sample clock period. Unit: seconds
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.sampleClockPeriod	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
FrIf	FrIf/FrIfConfig/FrIfCluster
BSW Parameter	**BSW Type**
FrIfGdStaticSlot	EcucIntegerParamDef
BSW Description	
Duration of a static slot [Macroticks]. Remark: Range 4-661 for FlexRay Protocol 2.1 Rev. A compliance.	
M2 Template	**M2 Description**
TPS_SYST	The duration of a slot in the static segment. Unit: macroticks
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.staticSlotDuration	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
FrIf	FrIf/FrIfConfig/FrIfCluster
BSW Parameter	**BSW Type**
FrIfGdSymbolWindow	EcucIntegerParamDef
BSW Description	
Duration of the symbol window [Macroticks]. Remark: Range 0-142 for FlexRay Protocol 2.1 Rev. A compliance.	
M2 Template	**M2 Description**
TPS_SYST	The duration of the symbol window. Unit: macroticks
M2 Parameter	

SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.transmissionStartSequenceDuration	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGdSymbolWindowActionPointOffset		EcucIntegerParamDef
BSW Description		
Number of macroticks the action point offset is from the beginning of the symbol window [Macroticks]. Remark: Set to GdActionPointOffset for FlexRay Protocol 2.1 Rev. A compliance.		
M2 Template	**M2 Description**	
TPS_SYST	Number of macroticks the action point offset is from the beginning of the symbol window [Macroticks].	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.symbolWindowActionPointOffset		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGdTSSTransmitter		EcucIntegerParamDef
BSW Description		
Number of bits in the Transmission Start Sequence [gdBits]. Remark: Lower limit 3 for FlexRay Protocol 2.1 Rev. A compliance.		
M2 Template	**M2 Description**	
TPS_SYST	Number of bits in the Transmission Start Sequence [gdBits].	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.transmissionStartSequenceDuration		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGdWakeupRxIdle		EcucIntegerParamDef
BSW Description		
Number of bits used by the node to test the duration of the 'idle' or HIGH phase of a received wakeup [gdBit]. Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolRxIdle. Lower limit 14 for FlexRay Protocol 2.1 Rev. A compliance.		
M2 Template	**M2 Description**	
TPS_SYST	Number of bits used by the node to test the duration of the 'idle' or HIGH phase of a received wakeup [gdBit].	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.wakeupRxIdle		
Mapping Rule		**Mapping Type**

1:1 mapping	full

BSW Module	**BSW Context**	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGdWakeupRxLow		EcucIntegerParamDef
BSW Description		
Number of bits used by the node to test the duration of the LOW phase of a received wakeup [gdBit]. Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolRxLow. Lower limit 11 for FlexRay Protocol 2.1 Rev. A compliance.		
M2 Template	**M2 Description**	
TPS_SYST	Number of bits used by the node to test the duration of the LOW phase of a received wakeup. Unit:bitDuration	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.wakeupRxLow		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGdWakeupRxWindow		EcucIntegerParamDef
BSW Description		
The size of the window used to detect wakeups [gdBit]. Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolRxWindow. Upper limit 301 for FlexRay Protocol 2.1 Rev. A compliance.		
M2 Template	**M2 Description**	
TPS_SYST	The size of the window used to detect wakeups [gdBit].	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.wakeupRxWindow		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
FrIf	FrIf/FrIfConfig/FrIfCluster	
BSW Parameter		**BSW Type**
FrIfGdWakeupTxActive		EcucIntegerParamDef
BSW Description		
Number of bits used by the node to transmit the LOW phase of awakeup symbol and the HIGH and LOW phases of a WUDOP [gdBit]. Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolTxLow.		
M2 Template	**M2 Description**	
TPS_SYST	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	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.wakeupTxActive		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context
FrIf	FrIf/FrIfConfig/FrIfCluster
BSW Parameter	**BSW Type**
FrIfGdWakeupTxIdle	EcucIntegerParamDef
BSW Description	
Number of bits used by the node to transmit the 'idle' part of a wakeup symbol [gdBit]. Remarks: This parameter maps to FlexRay Protocol 2.1 Rev. A parameter gdWakeupSymbolTxIdle.	
M2 Template	**M2 Description**
TPS_SYST	Number of bits used by the node to transmit the idle part of a wake up symbol. Unit: gDbit
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.wakeupTxIdle	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	BSW Context
FrIf	FrIf/FrIfConfig/FrIfCluster
BSW Parameter	**BSW Type**
FrIfJobList	EcucParamConfContainerDef
BSW Description	
This container specifies a list of all FlexRay Jobs of the Cluster to be performed by FrIf_JobListExec_<ClstIdx>().	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfJobList
BSW Parameter	**BSW Type**
FrIfAbsTimerRef	EcucSymbolicNameReferenceDef
BSW Description	
Reference to the absolute timer to be used to trigger the interrupt whose ISR contains the FrIf_JobListExec_<ClstIdx>() function.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfJobList
BSW Parameter	**BSW Type**
FrIfJob	EcucParamConfContainerDef

BSW Description	
A job may contain more than one operation that are executed at a specific point in time.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfJobList/FrIfJob
BSW Parameter	**BSW Type**
FrIfCommunicationOperation	EcucParamConfContainerDef
BSW Description	
A separate operation which is part of a FlexRay Job and defines what type of action is executed.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfJobList/FrIfJob/FrIfCommunicationOperation
BSW Parameter	**BSW Type**
FrIfCommunicationAction	EcucEnumerationParamDef
BSW Description	
The action to be performed in the FlexRay Operation	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfJobList/FrIfJob/FrIfCommunicationOperation
BSW Parameter	**BSW Type**
FrIfCommunicationOperationIdx	EcucIntegerParamDef
BSW Description	
For each FlexRay Communication Job, this index spans a range of zero-based consecutive values and thus defines the order of the FlexRay Communication Operation in the respective FlexRay Communication Job.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfJobList/FrIfJob/FrIfCommunicationOperation	
BSW Parameter		**BSW Type**
FrIfLPduIdxRef		EcucReferenceDef
BSW Description		
Reference to a L-PDu index		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfJobList/FrIfJob/FrIfCommunicationOperation	
BSW Parameter		**BSW Type**
FrIfRxComOpMaxLoop		EcucIntegerParamDef
BSW Description		
Defines the maximum number of loops for the receive RECEIVE_AND_INDICATE (Use case: emptying a FIFO). Please note that the parameter is mandatory if FrIfCommunicationAction parameter is set to RECEIVE_AND_INDICATE. For all other operations this parameter can be ignored.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfJobList/FrIfJob	
BSW Parameter		**BSW Type**
FrIfCycle		EcucIntegerParamDef
BSW Description		
The FlexRay Cycle in which the communication operation will execute this job		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfJobList/FrIfJob	
BSW Parameter		**BSW Type**
FrIfMacrotick		EcucIntegerParamDef
BSW Description		
Macrotick offset in the Cycle [Macrotick]		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context
FrIf	FrIf/FrIfConfig/FrIfCluster/FrIfJobList/FrIfJob
BSW Parameter	**BSW Type**
FrIfMaxIsrDelay	EcucIntegerParamDef
BSW Description	
The maximum delay in macroticks the FrIf_JoblistExec_<cluster>() function is processed after the absolute timer interrupt was triggered.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context
FrIf	FrIf/FrIfConfig/FrIfCluster
BSW Parameter	**BSW Type**
FrIfMainFunctionPeriod	EcucFloatParamDef
BSW Description	
The execution cycle of the FrIf_MainFunction_<cluster>() in seconds. The FrIf does not require this information but the BSW scheduler, which invokes the cluster main functions, needs it in order to plan its tasks.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context
FrIf	FrIf/FrIfConfig/FrIfCluster
BSW Parameter	**BSW Type**
FrIfSafetyMargin	EcucIntegerParamDef
BSW Description	
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.	
M2 Template	**M2 Description**
TPS_SYST	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.
M2 Parameter	
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.safetyMargin	
Mapping Rule	**Mapping Type**
1:1 mapping	full

<table>
<tr><th>BSW Module</th><th colspan="2">BSW Context</th></tr>
<tr><td>FrIf</td><td colspan="2">FrIf/FrIfConfig</td></tr>
<tr><th colspan="2">BSW Parameter</th><th>BSW Type</th></tr>
<tr><td colspan="2">FrIfFrameStructure</td><td>EcucParamConfContainerDef</td></tr>
<tr><th colspan="3">BSW Description</th></tr>
<tr><td colspan="3">The Frame structure specifies a Construction Plan how a Frame is assembled with PDUs and their respective Update-Bits.</td></tr>
<tr><th>M2 Template</th><th colspan="2">M2 Description</th></tr>
<tr><td>TPS_SYST</td><td colspan="2">Data frame which is sent over a communication medium. This element describes the pure Layout of a frame sent on a channel.</td></tr>
<tr><th colspan="3">M2 Parameter</th></tr>
<tr><td colspan="3">SystemTemplate::Fibex::FibexCore::CoreCommunication::Frame</td></tr>
<tr><th colspan="2">Mapping Rule</th><th>Mapping Type</th></tr>
<tr><td colspan="2">Create container for each FlexRay Frame that is transmitted or received by the regarded ECU. IPduToFrameMapping element in the System Template contains the construction plan.</td><td>full</td></tr>
</table>

<table>
<tr><th>BSW Module</th><th colspan="2">BSW Context</th></tr>
<tr><td>FrIf</td><td colspan="2">FrIf/FrIfConfig/FrIfFrameStructure</td></tr>
<tr><th colspan="2">BSW Parameter</th><th>BSW Type</th></tr>
<tr><td colspan="2">FrIfByteOrder</td><td>EcucEnumerationParamDef</td></tr>
<tr><th colspan="3">BSW Description</th></tr>
<tr><td colspan="3">This parameter defines the ByteOrder of all Pdus that are mapped into the Frame.

The absolute position of a Pdu in the Frame is
determined by the definition of the ByteOrder parameter:
If BIG_ENDIAN is specified, the FrIfPduOffset indicates the position of the most significant bit in the Frame.
If LITTLE_ENDIAN is specified, the FrIfPduOffset indicates the position of the least significant bit in the Frame.</td></tr>
<tr><th>M2 Template</th><th colspan="2">M2 Description</th></tr>
<tr><td>TPS_SYST</td><td colspan="2">This attribute defines the order of the bytes of the Pdu and the packing into the Frame. The byte ordering "Little Endian" (MostSignificantByteLast) and "Big Endian" (MostSignificantByteFirst) can be selected.</td></tr>
<tr><th colspan="3">M2 Parameter</th></tr>
<tr><td colspan="3">SystemTemplate::Fibex::FibexCore::CoreCommunication::PduToFrameMapping.packingByteOrder</td></tr>
<tr><th colspan="2">Mapping Rule</th><th>Mapping Type</th></tr>
<tr><td colspan="2">1:1 mapping</td><td>full</td></tr>
</table>

<table>
<tr><th>BSW Module</th><th colspan="2">BSW Context</th></tr>
<tr><td>FrIf</td><td colspan="2">FrIf/FrIfConfig/FrIfFrameStructure</td></tr>
<tr><th colspan="2">BSW Parameter</th><th>BSW Type</th></tr>
<tr><td colspan="2">FrIfPdusInFrame</td><td>EcucParamConfContainerDef</td></tr>
<tr><th colspan="3">BSW Description</th></tr>
<tr><td colspan="3">This container holds all the information about a PDU in a FlexRay Frame.</td></tr>
<tr><th>M2 Template</th><th colspan="2">M2 Description</th></tr>
<tr><td>TPS_SYST</td><td colspan="2">A PduToFrameMapping defines the composition of Pdus in each frame.</td></tr>
<tr><th colspan="3">M2 Parameter</th></tr>
<tr><td colspan="3">SystemTemplate::Fibex::FibexCore::CoreCommunication::PduToFrameMapping</td></tr>
<tr><th colspan="2">Mapping Rule</th><th>Mapping Type</th></tr>
<tr><td colspan="2">Container must be created for each IPduToFrameMapping element inside the frame.</td><td>full</td></tr>
</table>

BSW Module	**BSW Context**	
FrIf	FrIf/FrIfConfig/FrIfFrameStructure/FrIfPdusInFrame	
BSW Parameter		**BSW Type**
FrIfPduOffset		EcucIntegerParamDef
BSW Description		
The value specifies the offset of the PDU within the Frame [bytes].		
M2 Template	**M2 Description**	
TPS_SYST	This attribute describes the bitposition of a Pdu within a Frame. 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.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::PduToFrameMapping.startPosition		
Mapping Rule		**Mapping Type**
Please note that the startPosition attribute is defined in bits and the FrIfPdu Offset parameter is defined in bytes.		full

BSW Module	**BSW Context**	
FrIf	FrIf/FrIfConfig/FrIfFrameStructure/FrIfPdusInFrame	
BSW Parameter		**BSW Type**
FrIfPduRef		EcucReferenceDef
BSW Description		
This is the reference to the local definition of a PDU.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
FrIf	FrIf/FrIfConfig/FrIfFrameStructure/FrIfPdusInFrame	
BSW Parameter		**BSW Type**
FrIfPduUpdateBitOffset		EcucIntegerParamDef
BSW Description		
This value specifies where the PDU's Update-Bit is stored in the Frame (bit location of PDU's Update-Bit in the FlexRay Frame).		
M2 Template	**M2 Description**	
TPS_SYST	Indication to the receivers that the corresponding I-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.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::PduToFrameMapping.updateIndicationBitPosition		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
FrIf	FrIf/FrIfConfig	
BSW Parameter		**BSW Type**

FrIfMaxPduCnt		EcucIntegerParamDef	
BSW Description			
Maximum number of Pdus. This parameter is needed only in case of post-build loadable implementation using static memory allocation.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
FrIf	FrIf/FrIfConfig		
BSW Parameter		**BSW Type**	
FrIfPdu		EcucParamConfContainerDef	
BSW Description			
Contains PDU information. A PDU may be either a transmission PDU or a reception PDU.			
M2 Template	**M2 Description**		
TPS_SYST	Collection of all Pdus that can be routed through a bus interface.		
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreCommunication::Pdu			
Mapping Rule			**Mapping Type**
The container must be created for each Pdu that is contained in a FlexRay Frame.			full

BSW Module	**BSW Context**		
FrIf	FrIf/FrIfConfig/FrIfPdu		
BSW Parameter		**BSW Type**	
FrIfPduDirection		EcucChoiceContainerDef	
BSW Description			
A PDU is either transmit or receive			
M2 Template	**M2 Description**		
TPS_SYST	Communication Direction of the Connector Port (input or output Port).		
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreTopology::CommConnectorPort.communicationDirection			
Mapping Rule			**Mapping Type**
The PduTriggering contains a reference to a IPduPort with the communication Direction.			full

BSW Module	**BSW Context**		
FrIf	FrIf/FrIfConfig/FrIfPdu/FrIfPduDirection		
BSW Parameter		**BSW Type**	
FrIfRxPdu		EcucParamConfContainerDef	
BSW Description			
Receive PDU			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

	local

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfPdu/FrIfPduDirection/FrIfRxPdu	
BSW Parameter		**BSW Type**
FrIfRxIndicationName		EcucFunctionNameDef
BSW Description		
This parameter defines the name of the <User_RxIndication>. This parameter depends on the parameter FRIF_USERRXINDICATION_UL. If FRIF_USERRXINDICATION_UL equals FR_TP, FR_NM, PDUR or XCP, the name of the <User_RxIndication> is fixed. If FRIF_USERRXINDICATION_UL equals CDD, the name of the <User_RxIndication> is selectable.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfPdu/FrIfPduDirection/FrIfRxPdu	
BSW Parameter		**BSW Type**
FrIfRxPduRef		EcucReferenceDef
BSW Description		
Reference to the external PDU definition.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfPdu/FrIfPduDirection/FrIfRxPdu	
BSW Parameter		**BSW Type**
FrIfUserRxIndicationUL		EcucEnumerationParamDef
BSW Description		
This parameter defines the upper layer (UL) module to which the indication of the successfully received FRIFRXPDU has to be routed via <User_RxIndication>. This <User_RxIndication> has to be invoked when the indication of the configured FRIFRXPDU will be received by a Rx indication event from the FR Driver module. If no upper layer (UL) module is configured, no <User_RxIndication> has to be called in case of a Rx indication event of the FRIFRXPDU from the FR Driver module.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context

FrIf	FrIf/FrIfConfig/FrIfPdu/FrIfPduDirection		
BSW Parameter		**BSW Type**	
FrIfTxPdu		EcucParamConfContainerDef	
BSW Description			
This container specifies transmission PDUs.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
FrIf	FrIf/FrIfConfig/FrIfPdu/FrIfPduDirection/FrIfTxPdu		
BSW Parameter		**BSW Type**	
FrIfConfirm		EcucBooleanParamDef	
BSW Description			
Defines whether the transmission of a PDU should be checked and confirmed to the PDU owning BSW module.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
FrIf	FrIf/FrIfConfig/FrIfPdu/FrIfPduDirection/FrIfTxPdu		
BSW Parameter		**BSW Type**	
FrIfCounterLimit		EcucIntegerParamDef	
BSW Description			
This value states the maximum number of indication of ready PDU data to the FrIf (i.e. maximum number of invocations of FrIf_Transmit) without an intermediate transmission of the PDU.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**	
FrIf	FrIf/FrIfConfig/FrIfPdu/FrIfPduDirection/FrIfTxPdu	
BSW Parameter		**BSW Type**
FrIfImmediate		EcucBooleanParamDef
BSW Description		
Defines whether the PDU is transmitted immediate or decoupled.		
M2 Template	**M2 Description**	
M2 Parameter		

Mapping Rule	Mapping Type
	local

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfPdu/FrIfPduDirection/FrIfTxPdu	
BSW Parameter		**BSW Type**
FrIfNoneMode		EcucBooleanParamDef
BSW Description		
Using the "None-Mode" which means that there is no API FrIf_Transmit call of the upper layer for this PDU.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfPdu/FrIfPduDirection/FrIfTxPdu	
BSW Parameter		**BSW Type**
FrIfTxConfirmationName		EcucFunctionNameDef
BSW Description		
This parameter defines the name of the <User_TxConfirmation>. This parameter depends on the parameter FrIfUserTxUL. If FrIfUserTxUL equals FR_TP, FR_AR_TP, FR_NM, PDUR or XCP, the name of the <User_TxConfirmation> is fixed. If FrIfUserTxUL equals CDD, the name of the <User_TxConfirmation> is selectable.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrIf	FrIf/FrIfConfig/FrIfPdu/FrIfPduDirection/FrIfTxPdu	
BSW Parameter		**BSW Type**
FrIfTxPduId		EcucIntegerParamDef
BSW Description		
The global PDU identifier, which has to be used by the upper layer BSW module. The identifier has to be zero based and consecutive.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context
FrIf	FrIf/FrIfConfig/FrIfPdu/FrIfPduDirection/FrIfTxPdu

BSW Parameter	BSW Type
FrIfTxPduRef	EcucReferenceDef
BSW Description	
Reference to the external PDU definition.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context
FrIf	FrIf/FrIfConfig/FrIfPdu/FrIfPduDirection/FrIfTxPdu
BSW Parameter	**BSW Type**
FrIfUserTriggerTransmitName	EcucFunctionNameDef
BSW Description	
This parameter defines the name of the <User_TriggerTransmit>. This parameter depends on the parameter FrIfUserTxUL. If FrIfUserTxUL equals FR_TP, FR_NM, PDUR or XCP, the name of the <User_TriggerTransmit> is fixed. If FrIfUserTxUL equals CDD, the name of the <User_TriggerTransmit> is selectable.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context
FrIf	FrIf/FrIfConfig/FrIfPdu/FrIfPduDirection/FrIfTxPdu
BSW Parameter	**BSW Type**
FrIfUserTxUL	EcucEnumerationParamDef
BSW Description	
This parameter defines the upper layer (UL) module to which the trigger of the Pdu to be transmitted (via the <User_TriggerTransmit>) or the confirmation of the successfully transmitted Pdu has to be routed (via the <User_TxConfirmation>). Please note that handle IDs which are used in callback functions are defined by the upper layer module.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context
FrIf	FrIf
BSW Parameter	**BSW Type**
FrIfGeneral	EcucParamConfContainerDef
BSW Description	
This container contains the general configuration parameters of the FlexRay Interface.	
M2 Template	**M2 Description**

TPS_SYST	The CommunicationCluster is the main element to describe the topological connection of communicating ECUs.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster		
Mapping Rule		**Mapping Type**
Container must be created if the ECU is connected to a FlexRay Cluster		full

BSW Module	**BSW Context**	
FrIf	FrIf/FrIfGeneral	
BSW Parameter		**BSW Type**
FrIfAbsTimerIdx		EcucIntegerParamDef
BSW Description		
Maximum number of supported absolute timers.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
FrIf	FrIf/FrIfGeneral	
BSW Parameter		**BSW Type**
FrIfAllSlotsSupport		EcucBooleanParamDef
BSW Description		
Configuration parameter to enable/disable FrIf support to enable/disable of switching from key-slot / single-slot mode to all slot mode.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
FrIf	FrIf/FrIfGeneral	
BSW Parameter		**BSW Type**
FrIfCancelTransmitSupport		EcucBooleanParamDef
BSW Description		
Configuration parameter to enable/disable FrIf support to request the cancellation of the I-PDU transmission to FrDrv.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**

FrIf	FrIf/FrIfGeneral		
BSW Parameter		**BSW Type**	
FrIfDevErrorDetect		EcucBooleanParamDef	
BSW Description			
Switches the Development Error Detection and Notification on or off true: Development Error Detection and Notification on false: Development Error Detection and Notification off			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
FrIf	FrIf/FrIfGeneral		
BSW Parameter		**BSW Type**	
FrIfDisableLPduSupport		EcucBooleanParamDef	
BSW Description			
Configuration parameter to enable/disable FrIf support to disables the hardware resource of a LPdu for transmission/reception.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
FrIf	FrIf/FrIfGeneral		
BSW Parameter		**BSW Type**	
FrIfDisableTransceiverBranchSupport		EcucBooleanParamDef	
BSW Description			
Configuration parameter to enable/disable FrIf support to disable branches of an active star.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**	
FrIf	FrIf/FrIfGeneral	
BSW Parameter		**BSW Type**
FrIfEnableTransceiverBranchSupport		EcucBooleanParamDef
BSW Description		
Configuration parameter to enable/disable FrIf support to enable branches of an active star.		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**	
FrIf	FrIf/FrIfGeneral	
BSW Parameter		**BSW Type**
FrIfFreeOpAApiName		EcucStringParamDef
BSW Description		
API name that is called when FREE_OP_A is selected as communication operation. See also chapter 8.8.3 Configurable Interfaces.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
FrIf	FrIf/FrIfGeneral	
BSW Parameter		**BSW Type**
FrIfFreeOpBApiName		EcucStringParamDef
BSW Description		
API name that is called when FREE_OP_B is selected as communication operation. See also chapter 8.8.3 Configurable Interfaces.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
FrIf	FrIf/FrIfGeneral	
BSW Parameter		**BSW Type**
FrIfFreeOpsHeader		EcucStringParamDef
BSW Description		
Defines header file for configurable FREE_OP_A / FREE_OP_B functions.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
FrIf	FrIf/FrIfGeneral	
BSW Parameter		**BSW Type**

FrIfGetClockCorrectionSupport		EcucBooleanParamDef
BSW Description		
Configuration parameter to enable/disable FrIf support to enable/disable of polling the FlexRay Driver to getting CC clock correction values.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
FrIf	FrIf/FrIfGeneral	
BSW Parameter		**BSW Type**
FrIfGetGetChannelStatusSupport		EcucBooleanParamDef
BSW Description		
Configuration parameter to enable/disable FrIf support to enable/disable of polling the FlexRay Driver to getting error information about the FlexRay communications bus.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
FrIf	FrIf/FrIfGeneral	
BSW Parameter		**BSW Type**
FrIfGetNmVectorSupport		EcucBooleanParamDef
BSW Description		
Configuration parameter to enable/disable FrIf support to request the FlexRay hardware NMVector.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
FrIf	FrIf/FrIfGeneral	
BSW Parameter		**BSW Type**
FrIfGetNumOfStartupFramesSupport		EcucBooleanParamDef
BSW Description		
Configuration parameter to enable/disable FrIf support to enable/disable of polling the FlexRay Driver for the actual number of received startup frames on the bus.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

	local

BSW Module	BSW Context	
FrIf	FrIf/FrIfGeneral	
BSW Parameter		**BSW Type**
FrIfGetSyncFrameListSupport		EcucBooleanParamDef
BSW Description		
Configuration parameter to enable/disable FrIf support to enable/disable of polling the FlexRay Driver to getting a list of actual received sync frames.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrIf	FrIf/FrIfGeneral	
BSW Parameter		**BSW Type**
FrIfGetTransceiverErrorSupport		EcucBooleanParamDef
BSW Description		
Configuration parameter to enable/disable FrIf support to get the FlexRay Transceiver errors by calling the FlexRay Transceiver module.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrIf	FrIf/FrIfGeneral	
BSW Parameter		**BSW Type**
FrIfGetWakeupRxStatusSupport		EcucBooleanParamDef
BSW Description		
Configuration parameter to enable/disable FrIf support to get the wakeup received information from the FlexRay controller.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrIf	FrIf/FrIfGeneral	
BSW Parameter		**BSW Type**
FrIfNumClstSupported		EcucIntegerParamDef
BSW Description		

Maximum number of FlexRay Clusters that the FlexRay Interface supports.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
FrIf	FrIf/FrIfGeneral	
BSW Parameter		**BSW Type**
FrIfNumCtrlSupported		EcucIntegerParamDef
BSW Description		
Maximum number of FlexRay CCs that the FlexRay Interface supports		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
FrIf	FrIf/FrIfGeneral	
BSW Parameter		**BSW Type**
FrIfPublicCddHeaderFile		EcucStringParamDef
BSW Description		
Defines header files for callback functions which shall be included in case of CDDs. Range of characters is 1.. 32.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
FrIf	FrIf/FrIfGeneral	
BSW Parameter		**BSW Type**
FrIfReadCCConfigApi		EcucBooleanParamDef
BSW Description		
Configuration parameter to enable/disable the optional FrIf_ReadCCConfig API.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**

FrIf	FrIf/FrIfGeneral		
BSW Parameter		**BSW Type**	
FrIfReconfigLPduSupport		EcucBooleanParamDef	
BSW Description			
Configuration parameter to enable/disable FrIf support to enable/disable the reconfiguration of a given LPdu according to the parameters (FrameId, Channel, CycleRepetition, CycleOffset, PayloadLength, HeaderCRC) at runtime.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
FrIf	FrIf/FrIfGeneral		
BSW Parameter		**BSW Type**	
FrIfUnusedBitValue		EcucIntegerParamDef	
BSW Description			
Set unused bits to a defined value.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
FrIf	FrIf/FrIfGeneral		
BSW Parameter		**BSW Type**	
FrIfVersionInfoApi		EcucBooleanParamDef	
BSW Description			
Enables/disables the existence of the FrIf_GetVersionInfo() API service true: FrIf_GetVersionInfo() API service exists false: FrIf_GetVersionInfo() API service does not exist			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

D.4.3 FrNm Mapping

BSW Module	**BSW Context**	
FrNm	FrNm	
BSW Parameter		**BSW Type**
FrNmChannelConfig		EcucParamConfContainerDef
BSW Description		

This container contains the configuration parameters for all FlexRay NM channels.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
FrNm	FrNm/FrNmChannelConfig	
BSW Parameter		**BSW Type**
FrNmChannel		EcucParamConfContainerDef
BSW Description		
This container contains the configuration parameters for a FlexRay NM Channel.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel	
BSW Parameter		**BSW Type**
FrNmChannelIdentifiers		EcucParamConfContainerDef
BSW Description		
This container contains instance specific identifiers related to the respective FlexRay Channel.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers	
BSW Parameter		**BSW Type**
FrNmActiveWakeupBitEnabled		EcucBooleanParamDef
BSW Description		
Enables/Disables the handling of the Active Wakeup Bit in the FrNm module.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers

BSW Parameter		BSW Type
FrNmCarWakeUpBitPosition		EcucIntegerParamDef
BSW Description		
Specifies the Bit position of the CWU within the NM-Message.		
M2 Template	**M2 Description**	
TPS_SYST	Specifies the bit position of the CarWakeUp within the NM-Message.	
M2 Parameter		
SystemTemplate::NetworkManagement::FlexrayNmCluster.nmCarWakeUpBitPosition		
Mapping Rule		**Mapping Type**
The position of the Car Wakeup bit in the Ecuc is defined by the configuration parameters FrNmCarWakeUpBytePosition and FrNmCarWakeUpBitPosition (position in wakeUpByte). In the SysT the position is described only by the bit position in the NmMessage.		full

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers	
BSW Parameter		**BSW Type**
FrNmCarWakeUpBytePosition		EcucIntegerParamDef
BSW Description		
Specifies the Byte position of the CWU within the NM-Message.		
M2 Template	**M2 Description**	
TPS_SYST	Specifies the bit position of the CarWakeUp within the NM-Message.	
M2 Parameter		
SystemTemplate::NetworkManagement::FlexrayNmCluster.nmCarWakeUpBitPosition		
Mapping Rule		**Mapping Type**
The position of the Car Wakeup bit in the Ecuc is defined by the configuration parameters FrNmCarWakeUpBytePosition and FrNmCarWakeUpBitPosition (position in wakeUpByte). In the SysT the position is described only by the bit position in the NmMessage.		full

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers	
BSW Parameter		**BSW Type**
FrNmCarWakeUpFilterEnabled		EcucBooleanParamDef
BSW Description		
If CWU filtering is supported, only the CWU bit within the NM message with source node identifier FrNmCarWakeUpFilterNodeId is considered as CWU request. FALSE - CWU Filtering is not supported TRUE - CWU Filtering is supported		
M2 Template	**M2 Description**	
TPS_SYST	If this attribute is set to true the CareWakeUp filtering is supported. In this case only the CarWakeUp bit within the NM message with source node identifier nmCarWakeUpFilterNodeId is considered as CarWakeUp request.	
M2 Parameter		
SystemTemplate::NetworkManagement::FlexrayNmCluster.nmCarWakeUpFilterEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers	
BSW Parameter		**BSW Type**

FrNmCarWakeUpFilterNodeId		EcucIntegerParamDef	
BSW Description			
Source node identifier for CWU filtering. If CWU filtering is supported, only the CWU bit within the NM message with source node identifier FrNmCarWakeUpFilterNodeId is considered as CWU request.			
M2 Template	**M2 Description**		
TPS_SYST	Source node identifier for CarWakeUp filtering. If CarWakeUp filtering is supported (nmCarWakeUpFilterEnabled), only the CarWakeUp bit within the NM message with source node identifier nmCarWakeUpFilterNodeId is considered as CarWakeUp request.		
M2 Parameter			
SystemTemplate::NetworkManagement::FlexrayNmCluster.nmCarWakeUpFilterNodeId			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	**BSW Context**		
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers		
BSW Parameter		**BSW Type**	
FrNmCarWakeUpRxEnabled		EcucBooleanParamDef	
BSW Description			
Enables or disables support of CarWakeUp bit evaluation in received NM messages. FALSE - CarWakeUp not supported TRUE - CarWakeUp supported			
M2 Template	**M2 Description**		
TPS_SYST	If set to true this attribute enables the support of CarWakeUp bit evaluation in received NM messages.		
M2 Parameter			
SystemTemplate::NetworkManagement::FlexrayNmCluster.nmCarWakeUpRxEnabled			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	**BSW Context**		
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers		
BSW Parameter		**BSW Type**	
FrNmChannelHandle		EcucSymbolicNameReferenceDef	
BSW Description			
Channel identifier configured for the respective instance of the NM. The FrNmChannelHandle shall be encoded in the FrNmRxPduId parameter which is passed to FrNm_RxIndication() function called by the FrIf.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers	
BSW Parameter		**BSW Type**

FrNmComMNetworkHandleRef		EcucSymbolicNameReferenceDef
BSW Description		
This reference points to the unique channel defined by the ComMChannel and provides access to the unique channel index value in ComMChannelId.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers	
BSW Parameter		**BSW Type**
FrNmControlBitVectorActive		EcucBooleanParamDef
BSW Description		
This parameter is used to activate or deactivate the control bit vector support for a Fr Nm Channel.		
M2 Template	**M2 Description**	
TPS_SYST	Used to activate or deactivate the control bit vector support for a Fr Nm Channel.	
M2 Parameter		
SystemTemplate::NetworkManagement::FlexrayNmCluster.nmControlBitVectorActive		
Mapping Rule		**Mapping Type**
full		full

BSW Module	**BSW Context**	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers	
BSW Parameter		**BSW Type**
FrNmNodeId		EcucIntegerParamDef
BSW Description		
NM node identifier configured for the respective FlexRay Channel. It is used for identifying the respective NM node in the NM-cluster. It must be unique for each NM node within one NM cluster.		
M2 Template	**M2 Description**	
TPS_SYST	Node identifier of local NmNode. Must be unique in the NmCluster.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmNode.nmNodeId		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers	
BSW Parameter		**BSW Type**
FrNmPduScheduleVariant		EcucEnumerationParamDef
BSW Description		

This parameter defines the PDU scheduling variant that should be used for this channel. Option 1 NM-Vote and NM-Data in static segment (one PDU) Option 2 NM-Vote and NM-Data in dynamic segment (one PDU) Option 3 NM-Vote and NM-Data in static segment (separate PDU) Option 4 NM-Vote in static segment and NM-Data in dynamic segment Option 5 NM-Vote in dynamic segment and NM-Data in static segment Option 6 NM-Vote and NM-Data in dynamic segment (separate PDU) Option 7 Combined NM-Vote and CBV in static segment and NM-Data in dynamic segment		
M2 Template	**M2 Description**	
TPS_SYST	FrNm schedule variant according to FrNm SWS.	
M2 Parameter		
SystemTemplate::NetworkManagement::FlexrayNmClusterCoupling.nmScheduleVariant		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers	
BSW Parameter		**BSW Type**
FrNmPnEnabled		EcucBooleanParamDef
BSW Description		
Enables or disables support of partial networking. false: Partial networking Range not supported true: Partial networking supported		
M2 Template	**M2 Description**	
TPS_SYST	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.	
M2 Parameter		
SystemTemplate::PncMapping::PncMapping		
Mapping Rule		**Mapping Type**
For every EcuInstance that references an ISignalIPduGroup that is referenced by a PncMapping the FrNmPnEnabled shall be set to true.		full

BSW Module	**BSW Context**	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers	
BSW Parameter		**BSW Type**
FrNmPnEraCalcEnabled		EcucBooleanParamDef
BSW Description		
Specifies if FrNm calculates the PN request information for external requests. (ERA) false: PN request are not calculated true: PN request are calculated		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers	
BSW Parameter		**BSW Type**
FrNmPnEraRxNSduRef		EcucReferenceDef
BSW Description		
Reference to a Pdu in the COM-Stack. Only one SduRef is required for FrNm because the EIRA is the aggregation over all FlexRay Channels.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers	
BSW Parameter		**BSW Type**
FrNmRepeatMessageBitActive		EcucBooleanParamDef
BSW Description		
This parameter is deprecated and will be removed in future. Old description: This parameter is used to activate or deactivate the repeat message bit support for a Fr Nm Channel.		
M2 Template	**M2 Description**	
TPS_SYST	Used to activate or deactivate the repeat message bit support for a Fr Nm Channel.	
M2 Parameter		
SystemTemplate::NetworkManagement::FlexrayNmCluster.nmRepeatMessageBitActive		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers	
BSW Parameter		**BSW Type**
FrNmRxPdu		EcucParamConfContainerDef
BSW Description		
This container describes the FlexRay NM RX PDU:s.		
M2 Template	**M2 Description**	
TPS_SYST	receive NM Pdu	
M2 Parameter		
SystemTemplate::NetworkManagement::NmNode.rxNmPdu		
Mapping Rule		**Mapping Type**
Create Container if the regarded NmNode recieves a Pdu		full

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers/FrNmRxPdu	
BSW Parameter		**BSW Type**
FrNmRxPduContainsData		EcucBooleanParamDef
BSW Description		

This parameter defines if the PDU contains NM Data.		
M2 Template	**M2 Description**	
TPS_SYST	Defines if the PDU contains NM Data. If the NmPdu does not aggregate any ISignalToIPduMappings it still may contain UserData that is set via Nm_SetUserData(). If the ISignalToIPduMapping exists then the nmDataInformation attribute shall be ignored.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::NmPdu.nmDataInformation,SystemTemplate::Fibex::FibexCore::CoreCommunication::NmPdu.iSignalToIPduMapping		
Mapping Rule		**Mapping Type**
Set to true if either the NmPdu aggregates one or more iSignalToIPduMappings, or - if none are aggregated - if nmDataInformation is true. Set to false in all other cases		full

BSW Module	**BSW Context**	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers/FrNmRxPdu	
BSW Parameter		**BSW Type**
FrNmRxPduContainsVote		EcucBooleanParamDef
BSW Description		
This parameter defines if the PDU contains NM Vote information.		
M2 Template	**M2 Description**	
TPS_SYST	efines if the PDU contains NM Vote information.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::NmPdu.nmVoteInformation		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers/FrNmRxPdu	
BSW Parameter		**BSW Type**
FrNmRxPduId		EcucIntegerParamDef
BSW Description		
PDU identifier configured for the respective FlexRay Channel. It is used for referring to the FlexRay Interface receive function. It must be consistent with the value configured in the FlexRay Interface. This ID is used for the combined reception of NM Vote and NM Data or for the reception of the NM Vote if NM Data is received in a separate PDU. ImplementationType: PduIdType		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers/FrNmRxPdu	
BSW Parameter		**BSW Type**
FrNmRxPduRef		EcucReferenceDef

BSW Description	
The reference to a PDU in the global PDU structure described in the AUTOSAR ECU Configuration Specification. This reference will be used by the FrIf module to derive the PDU Id.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers	
BSW Parameter		**BSW Type**
FrNmSynchronizationPointEnabled		EcucBooleanParamDef
BSW Description		
This parameter defines if this channel shall provide the synchronization point indication to the NM Interface.		
M2 Template	**M2 Description**	
TPS_SYST	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.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmCluster.nmSynchronizingNetwork		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers	
BSW Parameter		**BSW Type**
FrNmTxPdu		EcucParamConfContainerDef
BSW Description		
This container describes the FlexRay NM TX PDU:s.		
M2 Template	**M2 Description**	
TPS_SYST	transmit NM Pdu	
M2 Parameter		
SystemTemplate::NetworkManagement::NmNode.txNmPdu		
Mapping Rule		**Mapping Type**
Create Container if the regarded NmNode transmits a Pdu		full

BSW Module	**BSW Context**	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers/FrNmTxPdu	
BSW Parameter		**BSW Type**
FrNmTxConfirmationPduId		EcucIntegerParamDef
BSW Description		
Handle Id used by the Lower Layer when calling FrNm_TriggerTransmit() or FrNm_TxConfirmation().		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

	local

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers/FrNmTxPdu	
BSW Parameter		**BSW Type**
FrNmTxPduContainsData		EcucBooleanParamDef
BSW Description		
This parameted defines if the PDU contains NM Data.		
M2 Template	**M2 Description**	
TPS_SYST	Defines if the PDU contains NM Data. If the NmPdu does not aggregate any ISignalToIPduMappings it still may contain UserData that is set via Nm_SetUserData(). If the ISignalToIPduMapping exists then the nmDataInformation attribute shall be ignored.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::NmPdu.nmDataInformation,SystemTemplate::Fibex::FibexCore::CoreCommunication::NmPdu.iSignalToIPduMapping		
Mapping Rule		**Mapping Type**
Set to true if either the NmPdu aggregates one or more iSignalToIPduMappings, or - if none are aggregated - if nmDataInformation is true. Set to false in all other cases		full

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers/FrNmTxPdu	
BSW Parameter		**BSW Type**
FrNmTxPduContainsVote		EcucBooleanParamDef
BSW Description		
This parameted defines if the PDU contains NM Vote information.		
M2 Template	**M2 Description**	
TPS_SYST	Defines if the PDU contains NM Vote information.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::NmPdu.nmVoteInformation		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers/FrNmTxPdu	
BSW Parameter		**BSW Type**
FrNmTxPduRef		EcucReferenceDef
BSW Description		
The reference to a PDU in the global PDU structure described in the AUTOSAR ECU Configuration Specification. This reference is used to derive the PDU Id that is defined by the FrIf module.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context

FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers	
BSW Parameter		**BSW Type**
FrNmUserDataTxPdu		EcucParamConfContainerDef
BSW Description		
This optional container is used to configure the UserNm PDU. This container is only available if FrNmComUserDataSupport is enabled.		
M2 Template	**M2 Description**	
TPS_SYST	This optional aggregation is used to describe NmUserData 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.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::NmPdu.iSignalToIPduMapping		
Mapping Rule		**Mapping Type**
Create container for each NmPdu that aggregates the ISignalToIPduMapping element. The configuration for these Pdus (e.g. Transfer Properties) shall be derived from this information.		full

BSW Module	**BSW Context**	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers/FrNmUserDataTxPdu	
BSW Parameter		**BSW Type**
FrNmTxUserDataPduId		EcucIntegerParamDef
BSW Description		
This parameter defines the Handle ID of the NM User Data I-PDU.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelIdentifiers/FrNmUserDataTxPdu	
BSW Parameter		**BSW Type**
FrNmTxUserDataPduRef		EcucReferenceDef
BSW Description		
Reference to the NM User Data I-PDU in the global PDU collection.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel	
BSW Parameter		**BSW Type**
FrNmChannelTiming		EcucParamConfContainerDef
BSW Description		

This container contains instance-specific timing related to the respective FlexRay Channel.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelTiming	
BSW Parameter		**BSW Type**
FrNmDataCycle		EcucEnumerationParamDef
BSW Description		
Number of FlexRay Schedule Cycles needed to transmit the NM Data of all ECUs on the FlexRay bus		
M2 Template	**M2 Description**	
TPS_SYST	Number of FlexRay Communication Cycles needed to transmit the Nm Data PDUs of all FlexRay Nm Ecus of this FlexRayNmCluster.	
M2 Parameter		
SystemTemplate::NetworkManagement::FlexrayNmCluster.nmDataCycle		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelTiming	
BSW Parameter		**BSW Type**
FrNmMainFunctionPeriod		EcucFloatParamDef
BSW Description		
This parameter defines the processing cycle of the main function of FrNm module in seconds.		
M2 Template	**M2 Description**	
TPS_SYST	Defines the processing cycle of the main function of FrNm module.	
M2 Parameter		
SystemTemplate::NetworkManagement::FlexrayNmCluster.nmMainFunctionPeriod		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelTiming	
BSW Parameter		**BSW Type**
FrNmMsgTimeoutTime		EcucFloatParamDef
BSW Description		
Timeout of a NM-message. It determines in seconds how long the NM shall wait with notification of transmission failure while communication errors occur on the bus.		
M2 Template	**M2 Description**	
TPS_SYST	Timeout of a NM message in seconds. It determines how long the NM shall wait with notification of transmission failure while communication errors occur on the bus.	
M2 Parameter		
SystemTemplate::NetworkManagement::FlexrayNmCluster.nmMessageTimeoutTime		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**		
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelTiming		
BSW Parameter		**BSW Type**	
FrNmReadySleepCnt		EcucIntegerParamDef	
BSW Description			
FrNm switches to bus sleep mode at the end of the FrNmReadySleepCnt+1 repetition cycle without any NM vote. E.g. on a value of "1", the NM-State Machine will leave the Ready Sleep State after two NM Repetition Cycles with no "keep awake" votes.			
M2 Template	**M2 Description**		
TPS_SYST	The value of this attribute influences the shutdown behavior of the FlexRay NM. FrNm switches to bus sleep mode nmReadySleepTime seconds after the completion of the last repetition cycle containing a NM vote.		
M2 Parameter			
SystemTemplate::NetworkManagement::FlexrayNmCluster.nmReadySleepCount			
Mapping Rule			**Mapping Type**
FrNmReadySleepCount = ((Float2Int(nmReadySleepTime/cycle))/nmRepetitionCycle)-1			full

BSW Module	**BSW Context**		
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelTiming		
BSW Parameter		**BSW Type**	
FrNmRemoteSleepIndTime		EcucFloatParamDef	
BSW Description			
Timeout for Remote Sleep Indication. It defines the time in seconds how long it shall take to recognize that all other nodes are ready to sleep. The value "0" denotes that no Remote Sleep Indication functionality is configured.			
M2 Template	**M2 Description**		
TPS_SYST	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.		
M2 Parameter			
SystemTemplate::NetworkManagement::FlexrayNmCluster.nmRemoteSleepIndicationTime			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	**BSW Context**		
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelTiming		
BSW Parameter		**BSW Type**	
FrNmRepeatMessageTime		EcucFloatParamDef	
BSW Description			
Timeout for Repeat Message State. Defines the time in seconds how long the NM shall stay in the Repeat Message State. The value "0" denotes that no Repeat Message State is configured, which means that Repeat Message State is transient and implies that it is left immediately after entry and consequently no startup stability is guaranteed and no node detection procedure is possible.			
M2 Template	**M2 Description**		
TPS_SYST	Timeout for Repeat Message State. Defines the time in seconds how long the NM shall stay in the Repeat Message State.		
M2 Parameter			
SystemTemplate::NetworkManagement::FlexrayNmCluster.nmRepeatMessageTime			
Mapping Rule			**Mapping Type**

1:1 mapping	full

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelTiming	
BSW Parameter		**BSW Type**
FrNmRepetitionCycle		EcucEnumerationParamDef
BSW Description		
Number of Flexray Schedule Cycles used to repeat the transmission of the Nm vote of all ECUs on the Flexray Bus.		
M2 Template	**M2 Description**	
TPS_SYST	Number of FlexRay Communication Cycles used to repeat the transmission of the Nm vote PDUs of all FlexRay NmEcus of this FlexRayNmCluster. This value must be an integral multiple of nmVotingCycle.	
M2 Parameter		
SystemTemplate::NetworkManagement::FlexrayNmCluster.nmRepetitionCycle		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelTiming	
BSW Parameter		**BSW Type**
FrNmSyncLossTimer		EcucFloatParamDef
BSW Description		
This parameter is deprecated and will be removed in future. Old description: Initial value for the SyncLossTimer in seconds.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelTiming	
BSW Parameter		**BSW Type**
FrNmVoteInhibitionEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling the inhibition of vote changes from the next-to-last repetition cycle to the last repetition cycle before the Ready Sleep Counter expires.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context

FrNm	FrNm/FrNmChannelConfig/FrNmChannel/FrNmChannelTiming
BSW Parameter	**BSW Type**
FrNmVotingCycle	EcucEnumerationParamDef
BSW Description	
Number of FlexRay Schedule Cycles needed to transmit the Nm vote of all ECUs on the FlexRay Bus.	
M2 Template	**M2 Description**
TPS_SYST	Number of FlexRay CommunicationCycles needed to transmit the Nm vote of Pdus of all FlexRay NmEcus of this FlexRayNmCluster.
M2 Parameter	
SystemTemplate::NetworkManagement::FlexrayNmCluster.nmVotingCycle	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
FrNm	FrNm
BSW Parameter	**BSW Type**
FrNmGlobalConfig	EcucParamConfContainerDef
BSW Description	
This container contains all global configuration parameters for the FrNm module.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
FrNm	FrNm/FrNmGlobalConfig
BSW Parameter	**BSW Type**
FrNmGlobalConstants	EcucParamConfContainerDef
BSW Description	
This container contains module constants related to the FlexRay NM functionality.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalConstants
BSW Parameter	**BSW Type**
FrNmNumberOfClusters	EcucIntegerParamDef
BSW Description	
Number of AUTOSAR FR NM clusters allowed within one ECU.	
M2 Template	**M2 Description**
M2 Parameter	

Mapping Rule	Mapping Type
	local

BSW Module	BSW Context	
FrNm	FrNm/FrNmGlobalConfig	
BSW Parameter		**BSW Type**
FrNmGlobalFeatures		EcucParamConfContainerDef
BSW Description		
This container contains module features related to the FlexRay NM functionality.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures	
BSW Parameter		**BSW Type**
FrNmBusSynchronizationEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling the bus synchronization.		
M2 Template	**M2 Description**	
TPS_SYST	Enables bus synchronization support.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmBusSynchronizationEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures	
BSW Parameter		**BSW Type**
FrNmComUserDataSupport		EcucBooleanParamDef
BSW Description		
Enable/disable the user data support.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		full

BSW Module	BSW Context	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures	
BSW Parameter		**BSW Type**
FrNmControlBitVectorEnabled		EcucBooleanParamDef
BSW Description		

Pre-processor switch for enabling control bit vector support. calculationFormula = If (FrNmNodeDetectionEnabled == False) then Equal(False) else Equal(False or True)	
M2 Template	**M2 Description**
TPS_SYST	Enables control bit vector support.
M2 Parameter	
SystemTemplate::NetworkManagement::FlexrayNmClusterCoupling.nmControlBitVectorEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures	
BSW Parameter		**BSW Type**
FrNmCoordinatorSyncSupport		EcucBooleanParamDef
BSW Description		
Enables/disables the coordinator synchronization support.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures	
BSW Parameter		**BSW Type**
FrNmCycleCounterEmulation		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling the cycle counter emulation.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures	
BSW Parameter		**BSW Type**
FrNmDualChannelPduEnable		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling the support of dual channel transmission and reception of NM messages.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures	
BSW Parameter		**BSW Type**
FrNmHwVoteEnable		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling the processing of FlexRay Hardware aggregated NM-Votes. This switch enables/disables the optional API FrIf_GetNmVector.		
M2 Template	**M2 Description**	
TPS_SYST	Switch for enabling the processing of FlexRay Hardware aggregated NM-Votes.	
M2 Parameter		
SystemTemplate::NetworkManagement::FlexrayNmEcu.nmHwVoteEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures	
BSW Parameter		**BSW Type**
FrNmNodeDetectionEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling node detection support. calculationFormula = If (FrNmPassiveModeEnabled == False) then Equal(NmNodeDetectionEnabled) else Equal(False)		
M2 Template	**M2 Description**	
TPS_SYST	Enables the Request Repeat Message Request support. Only valid if nmNodeIdEnabled is set to true.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmNodeDetectionEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures	
BSW Parameter		**BSW Type**
FrNmPassiveModeEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling Passive Node Configuration support. calculationFormula = Equal(NmPassiveModeEnabled)		
M2 Template	**M2 Description**	
TPS_SYST	Enables support of the Passive Mode.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmPassiveModeEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures	
BSW Parameter		**BSW Type**
FrNmPduRxIndicationEnabled		EcucBooleanParamDef

BSW Description		
Pre-processor switch for enabling PDU reception indication.		
M2 Template	**M2 Description**	
TPS_SYST	Switch for enabling the PDU Rx Indication.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmPduRxIndicationEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context		
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures		
BSW Parameter		**BSW Type**	
FrNmPnEiraCalcEnabled		EcucBooleanParamDef	
BSW Description			
Specifies if FrNm calculates the PN request information for internal an external requests. (EIRA) true: PN request are calculated false: PN request are not calculated			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures		
BSW Parameter		**BSW Type**	
FrNmPnEiraRxNSduRef		EcucReferenceDef	
BSW Description			
Reference to a Pdu in the COM-Stack. Only one SduRef is required for FrNm because the EIRA is the aggregation over all FlexRay Channels.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures		
BSW Parameter		**BSW Type**	
FrNmPnInfo		EcucParamConfContainerDef	
BSW Description			
PN information configuration			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

	local

BSW Module	BSW Context	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures/FrNmPnInfo	
BSW Parameter		**BSW Type**
FrNmPnFilterMaskByte		EcucParamConfContainerDef
BSW Description		
Filter mask byte configuration		
M2 Template	**M2 Description**	
TPS_SYST	Bit mask for FlexRay Payload used to configure the FlexRay Transceiver for partial network wakeup. This attribute should not be computed from the pncIdentifier values in order to support future introduction of additional PNCs. Note that for one EcuInstance all contributing pncFilterDataMask will be bitwise ORed to obtain the value of FrNmPnFilterMaskByte. Note that this data mask is calculated over the whole payload (8 Byte) of the NmPdu ignoring the leading bytes which do not contain pncVector information. The number of leading bytes which shall be ignored is equivalent to the value of System.pncVectorOffset.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationConnector.pncFilterDataMask		
Mapping Rule		**Mapping Type**
For one EcuInstance all contributing pncFilterDataMask will be bitwise ORed to obtain aggregated pncFilterDataMask value for this ECU. Since the pncFilterDataMask is calculated over the whole payload (8 Byte) of the NmPdu, the leading Bytes of this aggregated pncFilterDataMask shall be ignored based on the System.pncVectorOffset value. In order to get the FrNmPnFilterMaskByteIndex and FrNmPnFilterMaskByteValue for all the bytes aggregated pncFilterDataMask shall be processed in a littleEndian way. E.g. if pncVectorOffset = 2 and aggregated pncFilterDataMask has the value 2ˆ 63 this will end up in a FrNmPnFilterMaskByte with FrNmPnFilterMaskByteIndex = 5 and FrNmPnFilterMaskByteValue = 128.		full

BSW Module	BSW Context	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures/FrNmPnInfo/FrNmPnFilterMaskByte	
BSW Parameter		**BSW Type**
FrNmPnFilterMaskByteIndex		EcucIntegerParamDef
BSW Description		
Index of the filter mask byte. Specifies the position within the filter mask byte array.		
M2 Template	**M2 Description**	

TPS_SYST	Bit mask for FlexRay Payload used to configure the FlexRay Transceiver for partial network wakeup. This attribute should not be computed from the pncIdentifier values in order to support future introduction of additional PNCs. Note that for one EcuInstance all contributing pncFilterDataMask will be bitwise ORed to obtain the value of FrNmPnFilterMaskByte. Note that this data mask is calculated over the whole payload (8 Byte) of the NmPdu ignoring the leading bytes which do not contain pncVector information. The number of leading bytes which shall be ignored is equivalent to the value of System.pncVectorOffset.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationConnector.pncFilterDataMask		
Mapping Rule		**Mapping Type**
For one EcuInstance all contributing pncFilterDataMask will be bitwise ORed to obtain aggregated pncFilterDataMask value for this ECU. Since the pncFilterDataMask is calculated over the whole payload (8 Byte) of the NmPdu, the leading Bytes of this aggregated pncFilterDataMask shall be ignored based on the System.pncVectorOffset value. In order to get the FrNmPnFilterMaskByteIndex and FrNmPnFilterMaskByteValue for all the bytes aggregated pncFilterDataMask shall be processed in a littleEndian way. E.g. if pncVectorOffset = 2 and aggregated pncFilterDataMask has the value 2ˆ 63 this will end up in a FrNmPnFilterMaskByte with FrNmPnFilterMaskByteIndex = 5 and FrNmPnFilterMaskByteValue = 128.		full

BSW Module	**BSW Context**	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures/FrNmPnInfo/FrNmPnFilterMaskByte	
BSW Parameter		**BSW Type**
FrNmPnFilterMaskByteValue		EcucIntegerParamDef
BSW Description		
Parameter to configure the filter mask byte.		
M2 Template	**M2 Description**	
TPS_SYST	Bit mask for FlexRay Payload used to configure the FlexRay Transceiver for partial network wakeup. This attribute should not be computed from the pncIdentifier values in order to support future introduction of additional PNCs. Note that for one EcuInstance all contributing pncFilterDataMask will be bitwise ORed to obtain the value of FrNmPnFilterMaskByte. Note that this data mask is calculated over the whole payload (8 Byte) of the NmPdu ignoring the leading bytes which do not contain pncVector information. The number of leading bytes which shall be ignored is equivalent to the value of System.pncVectorOffset.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCommunicationConnector.pncFilterDataMask		
Mapping Rule		**Mapping Type**

For one EcuInstance all contributing pncFilterDataMask will be bitwise ORed to obtain aggregated pncFilterDataMask value for this ECU. Since the pncFilterDataMask is calculated over the whole payload (8 Byte) of the NmPdu, the leading Bytes of this aggregated pncFilterDataMask shall be ignored based on the System.pncVectorOffset value. In order to get the FrNmPnFilterMaskByteIndex and FrNmPnFilterMaskByteValue for all the bytes aggregated pncFilterDataMask shall be processed in a littleEndian way. E.g. if pncVectorOffset = 2 and aggregated pncFilterDataMask has the value 2ˆ 63 this will end up in a FrNmPnFilterMaskByte with FrNmPnFilterMaskByteIndex = 5 and FrNmPnFilterMaskByteValue = 128.	full

BSW Module	**BSW Context**	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures/FrNmPnInfo	
BSW Parameter		**BSW Type**
FrNmPnInfoLength		EcucIntegerParamDef
BSW Description		
Specifies the length of the PN request information in the NM message.		
M2 Template	**M2 Description**	
TPS_SYST	Length of the partial networking request release information vector (in bytes).	
M2 Parameter		
SystemTemplate::System.pncVectorLength		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures/FrNmPnInfo	
BSW Parameter		**BSW Type**
FrNmPnInfoOffset		EcucIntegerParamDef
BSW Description		
Specifies the offset of the PN request information in the NM message.		
M2 Template	**M2 Description**	
TPS_SYST	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.	
M2 Parameter		
SystemTemplate::System.pncVectorOffset		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures	
BSW Parameter		**BSW Type**
FrNmPnResetTime		EcucFloatParamDef
BSW Description		
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. The value shall be the same for every channel. Thus it is a global config parameter.		
M2 Template	**M2 Description**	
M2 Parameter		

Mapping Rule	Mapping Type
	local

BSW Module	BSW Context	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures	
BSW Parameter		**BSW Type**
FrNmRemoteSleepIndicationEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling remote sleep indication. calculationFormula = If (FrNmPassiveModeEnabled == True) then Equal(False) else Equal(False or True)		
M2 Template	**M2 Description**	
TPS_SYST	Switch for enabling the PDU Rx Indication.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmRemoteSleepIndEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures	
BSW Parameter		**BSW Type**
FrNmRepeatMessageBitEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling the repeat message bit support.		
M2 Template	**M2 Description**	
TPS_SYST	Switch for enabling the Repeat Message Bit Indication.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmRepeatMsgIndEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures	
BSW Parameter		**BSW Type**
FrNmSourceNodeIdentifierEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling SourceNodeIdentifier support.		
M2 Template	**M2 Description**	
TPS_SYST	Enables the source node identifier.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmNodeIdEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures	
BSW Parameter		**BSW Type**
FrNmStateChangeIndicationEnabled		EcucBooleanParamDef

BSW Description		
Pre-processor switch for enabling state change indication.		
M2 Template	**M2 Description**	
TPS_SYST	Switch for enabling remote sleep indication support.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmStateChangeIndEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures	
BSW Parameter		**BSW Type**
FrNmUserDataEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling user data support.		
M2 Template	**M2 Description**	
TPS_SYST	Switch for enabling user data support.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmUserDataEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalFeatures	
BSW Parameter		**BSW Type**
FrNmVotingNextToLastRepetitionCycleDisable		EcucBooleanParamDef
BSW Description		
Pre-processor switch for disabling vote changes in the last two repetition cycles before the Ready Sleep Counter expires.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrNm	FrNm/FrNmGlobalConfig	
BSW Parameter		**BSW Type**
FrNmGlobalProperties		EcucParamConfContainerDef
BSW Description		
This container contains module properties related to the FlexRay NM functionality.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalProperties	
BSW Parameter		**BSW Type**
FrNmDevErrorDetect		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling development error detection		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalProperties	
BSW Parameter		**BSW Type**
FrNmMainAcrossFrCycle		EcucBooleanParamDef
BSW Description		
Parameter describing if the execution of FrNm_Main function crosses the FlexRay cycle boundary or not.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrNm	FrNm/FrNmGlobalConfig/FrNmGlobalProperties	
BSW Parameter		**BSW Type**
FrNmVersionInfoApi		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling version info API support.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

D.4.4 FrTp Mapping

BSW Module	BSW Context	
FrTp	FrTp	
BSW Parameter		**BSW Type**
FrTpGeneral		EcucParamConfContainerDef
BSW Description		
This container contains the general configuration parameters of the FlexRay Transport Protocol module.		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
FrTp	FrTp/FrTpGeneral
BSW Parameter	**BSW Type**
FrTpAckRt	EcucBooleanParamDef
BSW Description	
Preprocessor switch for enabling the Acknowledgement and retry mechanisms. True: Acknowledge and Retry is enabled False: Acknowledge and Retry is disabled	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
FrTp	FrTp/FrTpGeneral
BSW Parameter	**BSW Type**
FrTpChanNum	EcucIntegerParamDef
BSW Description	
Preprocessor switch for defining the number of concurrent channels the module supports. Up to 32 channels shall be definable here.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
FrTp	FrTp/FrTpGeneral
BSW Parameter	**BSW Type**
FrTpChangeParamApi	EcucBooleanParamDef
BSW Description	
Preprocessor switch for enabling the API to change FrTp communication parameters. True: ChangeParameter API is enabled False: ChangeParameter API is disabled.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context	
FrTp	FrTp/FrTpGeneral	
BSW Parameter		**BSW Type**
FrTpDevErrorDetect		EcucBooleanParamDef
BSW Description		
Preprocessor switch for enabling development error detection. True: Development Error Detection is enabled False: Development Error Detection is disabled		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrTp	FrTp/FrTpGeneral	
BSW Parameter		**BSW Type**
FrTpFullDuplexEnable		EcucBooleanParamDef
BSW Description		
Preprocessor switch for enabling full duplex mechanisms for all channels. True: Full duplex is enabled False: Fullduplex is disabled (Half duplex is enabled)		
M2 Template	**M2 Description**	
TPS_SYST	The full duplex mechanisms is enabled if this attribute is set to true. Otherwise half duplex is enabled.	
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayTpEcu.fullDuplexEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrTp	FrTp/FrTpGeneral	
BSW Parameter		**BSW Type**
FrTpMainFuncCycle		EcucFloatParamDef
BSW Description		
This parameter contains the calling period of the TPs Main Function. The parameter is specified in seconds.		
M2 Template	**M2 Description**	
TPS_SYST	The period between successive calls to the Main Function of the AUTOSAR TP. Specified in seconds.	
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayTpEcu.cycleTimeMainFunction		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrTp	FrTp/FrTpGeneral	
BSW Parameter		**BSW Type**

FrTpTransmitCancellation		EcucBooleanParamDef
BSW Description		
Preprocessor switch for enabling Transmit Cancellation and Receive Cancellation. True: Transmit/Receive Cancellation is enabled False: Transmit/Receive Cancellation is disabled		
M2 Template	**M2 Description**	
TPS_SYST	With this switch Tx and Rx Cancellation can be turned on or off.	
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayTpEcu.cancellation		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
FrTp	FrTp/FrTpGeneral	
BSW Parameter		**BSW Type**
FrTpUnknownMsgLength		EcucBooleanParamDef
BSW Description		
Preprocessor switch to support data transfer with unknown message length. True: Transmission with unknown message length is enabled False: Transmission with unknown message length is disabled		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
FrTp	FrTp/FrTpGeneral	
BSW Parameter		**BSW Type**
FrTpVersionInfoApi		EcucBooleanParamDef
BSW Description		
Preprocessor switch for enabling the Version info API. True: Version Info API is enabled False: Version Info API is disabled		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
FrTp	FrTp	
BSW Parameter		**BSW Type**
FrTpMultipleConfig		EcucParamConfContainerDef
BSW Description		
This container holds one or several multiple configuration sets.		

M2 Template	M2 Description	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context		
FrTp	FrTp/FrTpMultipleConfig		
BSW Parameter		**BSW Type**	
FrTpConnection		EcucParamConfContainerDef	
BSW Description			
This container contains the connection specific parameters to transfer N-PDUs via FlexRay TP.			
M2 Template	**M2 Description**		
TPS_SYST	A connection identifies the sender and the receiver of this particular communication. The FlexRayTp module routes a Pdu through this connection.		
M2 Parameter			
SystemTemplate::TransportProtocols::FlexrayTpConnection			
Mapping Rule			**Mapping Type**
Create container for each FlexRayTpConnection that is described in the ECU Extract.			full

BSW Module	BSW Context		
FrTp	FrTp/FrTpMultipleConfig/FrTpConnection		
BSW Parameter		**BSW Type**	
FrTpBandwidthLimitation		EcucBooleanParamDef	
BSW Description			
This parameter indicates whether the connection requires a bandwidth limitation or not. If FrTpBandwidthLimitation=True the sender shall send a StartFrame always on the first PDU of a PDU-Pool.			
M2 Template	**M2 Description**		
TPS_SYST	Specifies whether the connection requires a bandwidth limitation or not.		
M2 Parameter			
SystemTemplate::TransportProtocols::FlexrayTpConnection.bandwidthLimitation			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	BSW Context		
FrTp	FrTp/FrTpMultipleConfig/FrTpConnection		
BSW Parameter		**BSW Type**	
FrTpConCtrlRef		EcucReferenceDef	
BSW Description			
FrTpConnectionControlReference: This parameter defines a reference to a connection control container.			
M2 Template	**M2 Description**		
TPS_SYST	Reference to the connection control.		
M2 Parameter			
SystemTemplate::TransportProtocols::FlexrayTpConnection.tpConnectionControl			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	BSW Context	
FrTp	FrTp/FrTpMultipleConfig/FrTpConnection	
BSW Parameter		**BSW Type**
FrTpLa		EcucIntegerParamDef
BSW Description		
This parameter defines the Local Address for the respective connection. When the local instance is the sender, this is the Source Address within the TP frame. When the local instance is the receiver, this is the Target Address within the TP frame.		
M2 Template	**M2 Description**	
TPS_SYST	This parameter defines the Local Address for the respective connection. When the local instance is the sender, this is the Source Address within the TP frame. When the local instance is the receiver, this is the Target Address within the TP frame.	
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayTpConnection.transmitter,SystemTemplate::TransportProtocols::FlexrayTpConnection.receiver		
Mapping Rule		**Mapping Type**
If the local address is the sender it shall be derived from FlexrayTpConnection.transmitter. If the remote address is the receiver it shall be derived from FlexrayTpConnection.receiver.		full

BSW Module	BSW Context	
FrTp	FrTp/FrTpMultipleConfig/FrTpConnection	
BSW Parameter		**BSW Type**
FrTpMultipleReceiverCon		EcucBooleanParamDef
BSW Description		
This parameter defines, whether this connection is an 1:1 ('false') or an 1:n ('true') connection. If data segmentation is required this parameter is used to check whether segmentation is possible or not. If the connection is 1:n segmentation is not possible and an error will occur.		
M2 Template	**M2 Description**	
TPS_SYST	This parameter defines, whether this connection is an 1:1 ('false') or an 1:n ('true') connection.	
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayTpConnection.multicast		
Mapping Rule		**Mapping Type**
If FlexRayTpConnection contains a mulicast reference to TpAddress than set this parameter to true		full

BSW Module	BSW Context	
FrTp	FrTp/FrTpMultipleConfig/FrTpConnection	
BSW Parameter		**BSW Type**
FrTpRa		EcucIntegerParamDef
BSW Description		
This parameter defines the Remote Address for the respective connection. When the local instance is the sender, this is the Target Address within the TP frame. When the local instance is the receiver, this is the Source Address within the TP frame.		
M2 Template	**M2 Description**	
TPS_SYST	This parameter defines the Remote Address for the respective connection. When the local instance is the sender, this is the Target Address within the TP frame. When the local instance is the receiver, this is the Source Address within the TP frame.	
M2 Parameter		

SystemTemplate::TransportProtocols::FlexrayTpConnection.transmitter,SystemTemplate::TransportProtocols::FlexrayTpConnection.receiver	
Mapping Rule	**Mapping Type**
If the local address is the sender it shall be derived from FlexrayTpConnection.transmitter. If the remote address is the receiver it shall be derived from FlexrayTpConnection.receiver.	full

BSW Module	**BSW Context**	
FrTp	FrTp/FrTpMultipleConfig/FrTpConnection	
BSW Parameter		**BSW Type**
FrTpRxPduPoolRef		EcucReferenceDef
BSW Description		
This parameter defines a reference to a RxPduPool.		
M2 Template	**M2 Description**	
TPS_SYST	A connection has a reference to a set of NPdus (FrTpRxPduPool) which are defined for receiving data via this particular connection.	
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayTpConnection.rxPduPool		
Mapping Rule		**Mapping Type**
Create container if a rxPduPool is referenced by the FlexrayTpConnection.		full

BSW Module	**BSW Context**	
FrTp	FrTp/FrTpMultipleConfig/FrTpConnection	
BSW Parameter		**BSW Type**
FrTpRxSdu		EcucParamConfContainerDef
BSW Description		
This parameter defines the Rx Service Data Unit Identifier (Sdu Id) which uniquely identifies a data transfer (inter-module communication) between FrTp and PDUR.		
M2 Template	**M2 Description**	
TPS_SYST	Reference to the IPdu that is segmented by the Transport Protocol.	
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayTpConnection.directTpSdu		
Mapping Rule		**Mapping Type**
Create container if an Rx Pdu is referenced by the FlexRayTpConnection		full

BSW Module	**BSW Context**	
FrTp	FrTp/FrTpMultipleConfig/FrTpConnection/FrTpRxSdu	
BSW Parameter		**BSW Type**
FrTpRxSduId		EcucIntegerParamDef
BSW Description		
This unique identifier is used for change parameter request or receive cancellation from PduR to FrTp. ImplementationType: PduIdType		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**		
FrTp	FrTp/FrTpMultipleConfig/FrTpConnection/FrTpRxSdu		
BSW Parameter		**BSW Type**	
FrTpRxSduRef		EcucReferenceDef	
BSW Description			
Reference to a PDU in the global PDU structure.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
FrTp	FrTp/FrTpMultipleConfig/FrTpConnection		
BSW Parameter		**BSW Type**	
FrTpTxPduPoolRef		EcucReferenceDef	
BSW Description			
This parameter defines a reference to a TxPduPool.			
M2 Template	**M2 Description**		
TPS_SYST	A connection has a reference to a set of NPdus (FrTpTxPduPool) which are defined for sending data via this particular connection.		
M2 Parameter			
SystemTemplate::TransportProtocols::FlexrayTpConnection.txPduPool			
Mapping Rule			**Mapping Type**
Create reference if the FlexrayTpPduPool element is referenced by the Flexray TpConnection via the txPduPool reference.			full

BSW Module	**BSW Context**		
FrTp	FrTp/FrTpMultipleConfig/FrTpConnection		
BSW Parameter		**BSW Type**	
FrTpTxSdu		EcucParamConfContainerDef	
BSW Description			
This parameter defines the Tx Service Data Unit Identifier (Sdu Id) which uniquely identifies a data transfer (inter-module communication) between FrTp and PDUR.			
M2 Template	**M2 Description**		
TPS_SYST	Reference to the IPdu that is segmented by the Transport Protocol.		
M2 Parameter			
SystemTemplate::TransportProtocols::FlexrayTpConnection.directTpSdu			
Mapping Rule			**Mapping Type**
Create container if an Tx Pdu is referenced by the FlexRayTpConnection			full

BSW Module	**BSW Context**	
FrTp	FrTp/FrTpMultipleConfig/FrTpConnection/FrTpTxSdu	
BSW Parameter		**BSW Type**
FrTpTxSduId		EcucIntegerParamDef
BSW Description		
This is a unique identifier for a to be transmitted message from the PduR to the FrTp. ImplementationType: PduIdType		

M2 Template	M2 Description	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrTp	FrTp/FrTpMultipleConfig/FrTpConnection/FrTpTxSdu	
BSW Parameter		**BSW Type**
FrTpTxSduRef		EcucReferenceDef
BSW Description		
Reference to a PDU in the global PDU structure.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrTp	FrTp/FrTpMultipleConfig	
BSW Parameter		**BSW Type**
FrTpConnectionControl		EcucParamConfContainerDef
BSW Description		
This container contains the configuration parameters to control a FlexRay TP connection.		
M2 Template	**M2 Description**	
TPS_SYST	Configuration parameters to control a FlexRay TP connection.	
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayTpConnectionControl		
Mapping Rule		**Mapping Type**
Create container for each FlexRayTpConnectionControl that is described in the ECU Extract.		full

BSW Module	BSW Context	
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl	
BSW Parameter		**BSW Type**
FrTpAckType		EcucEnumerationParamDef
BSW Description		
This parameter defines the type of acknowledgement which is used for the specific channel.		
M2 Template	**M2 Description**	
TPS_SYST	This parameter defines the type of acknowledgement which is used for the specific channel.	
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayTpConnectionControl.ackType		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context

FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl
BSW Parameter	**BSW Type**
FrTpMaxAr	EcucIntegerParamDef
BSW Description	
This parameter defines the maximum number of trying to send a frame when a TIMEOUT AR occurs.	
M2 Template	**M2 Description**
TPS_SYST	This parameter defines the maximum number of trying to send a frame when a TIMEOUT AR occurs (depending on whether retry is configured).
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayTpConnectionControl.maxAr	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl
BSW Parameter	**BSW Type**
FrTpMaxAs	EcucIntegerParamDef
BSW Description	
This parameter defines the maximum number of trying to send a frame when a TIMEOUT AS occurs.	
M2 Template	**M2 Description**
TPS_SYST	This parameter defines the maximum number of trying to send a frame when a TIMEOUT AS occurs (depending on whether retry is configured)
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayTpConnectionControl.maxAs	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl
BSW Parameter	**BSW Type**
FrTpMaxBufReq	EcucIntegerParamDef
BSW Description	
This parameter is used to limit the number of retries for PduR_FrTpCopyTxData when no timer is active.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl
BSW Parameter	**BSW Type**
FrTpMaxBufferSize	EcucIntegerParamDef
BSW Description	
Limits the maximal buffer size the FrTp can choose in order to limit the amount of Tx buffer that will be requested at the sender side in a segmented transfer.	
M2 Template	**M2 Description**

TPS_SYST	This parameter is only relevant when having retry activated. It limits the maximal block size the FrTp can choose in order to limit the amount of Tx buffer that will be requested at the sender side in a segmented transfer.	
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayTpConnectionControl.maxBufferSize		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl	
BSW Parameter		**BSW Type**
FrTpMaxFCWait		EcucIntegerParamDef
BSW Description		
This parameter defines the maximum number of FlowControl N-PDUs with FlowState "WAIT"		
M2 Template	**M2 Description**	
TPS_SYST	This attribute defines the maximum number of FlowControl N-PDUs with FlowState "WAIT"	
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayTpConnectionControl.maxFcWait		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl	
BSW Parameter		**BSW Type**
FrTpMaxFrIf		EcucIntegerParamDef
BSW Description		
This parameter defines the maximum number of trying to send a frame when the FrIf returns an error.		
M2 Template	**M2 Description**	
TPS_SYST	This parameter defines the maximum number of trying to send a frame when the FrIf returns an error	
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayTpConnectionControl.maxFrIf		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl	
BSW Parameter		**BSW Type**
FrTpMaxNbrOfNPduPerCycle		EcucIntegerParamDef
BSW Description		
This parameter is part of the ISO 10681-2 protocol's FlowControl parameter "Bandwidth Control (BC)". It limits the number of N-Pdus the sender is allowed to transmit within a FlexRay cycle.		
M2 Template	**M2 Description**	
TPS_SYST	This parameter limits the number of N-Pdus the sender is allowed to transmit within a FlexRay cycle.	
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayTpConnectionControl.maxNumberOfNpduPerCycle		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl	
BSW Parameter		**BSW Type**
FrTpMaxRn		EcucIntegerParamDef
BSW Description		
This parameter defines the maximum number of retries (if retry is configured).		
M2 Template	**M2 Description**	
TPS_SYST	This parameter defines the maximum number of retries (if retry is configured for the particular channel).	
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayTpConnectionControl.maxRetries		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl	
BSW Parameter		**BSW Type**
FrTpSCexp		EcucIntegerParamDef
BSW Description		
This parameter is part of the ISO 10681-2 protocol's FlowControl parameter "Bandwidth Control (BC)". It represents the exponent to calculate the minimum number of "Separation Cycles" the sender has to wait for the next transmission of an FrTp N-Pdu.		
M2 Template	**M2 Description**	
TPS_SYST	Exponent to calculate the minimum number of "Separation Cycles" the sender has to wait for the next transmission of an FrTp N-Pdu.	
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayTpConnectionControl.separationCycleExponent		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl	
BSW Parameter		**BSW Type**
FrTpTimeBr		EcucFloatParamDef
BSW Description		
This parameter defines the time in seconds the FrTp requires to transmit a corresponding FlowControl Frame. According to ISO 10681-2 this parameter is a performance requirement.		
M2 Template	**M2 Description**	
TPS_SYST	Time (in seconds) until transmission of the next FlowControl N-PDU.	
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayTpConnectionControl.timeBr		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl	
BSW Parameter		**BSW Type**
FrTpTimeBuffer		EcucFloatParamDef
BSW Description		

This parameter is deprecated and will be removed in the future. Old description: This parameter defines the time in seconds of waiting for the next try to get a Tx or Rx buffer.	
M2 Template	**M2 Description**
TPS_SYST	This parameter defines the time of waiting for the next try to get a Tx or Rx buffer. Specified in seconds.
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayTpConnectionControl.timeBuffer	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl
BSW Parameter	**BSW Type**
FrTpTimeFrIf	EcucFloatParamDef
BSW Description	
This parameter defines the time in seconds of waiting for the next try (if retry is activated) to send via FrIf_Transmit.	
M2 Template	**M2 Description**
TPS_SYST	This parameter defines the time of waiting for the next try to send. Specified in seconds.
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayTpConnectionControl.timeFrIf	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl
BSW Parameter	**BSW Type**
FrTpTimeoutAr	EcucFloatParamDef
BSW Description	
This parameter states the timeout in seconds between the PDU transmit request of the Transport Layer to the FlexRay Interface and the corresponding confirmation of the FlexRay Interface on the receiver side (for FC or AF).	
M2 Template	**M2 Description**
TPS_SYST	This parameter states the timeout between the PDU transmit request of the Transport Layer to the FlexRay Interface and the corresponding confirmation of the FlexRay Interface on the receiver side (for FC or AF). Specified in seconds.
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayTpConnectionControl.timeoutAr	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl
BSW Parameter	**BSW Type**
FrTpTimeoutAs	EcucFloatParamDef
BSW Description	
This parameter specifies the timeout in seconds the FrIf shall confirm a transmitted Pdu to the FrTp.	
M2 Template	**M2 Description**

TPS_SYST	This attribute states the timeout between the PDU transmit request for the first PDU of the group used in the current connection of the Transport Layer to the FlexRay Interface and the corresponding confirmation of the FlexRay Interface
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayTpConnectionControl.timeoutAs	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl
BSW Parameter	**BSW Type**
FrTpTimeoutBs	EcucFloatParamDef
BSW Description	
This parameter defines the timeout in seconds for waiting for an FC or AF on the sender side in a 1:1 connection.	
M2 Template	**M2 Description**
TPS_SYST	This parameter defines the timeout in seconds for waiting for an FC or AF on the sender side in a 1:1 connection.
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayTpConnectionControl.timeoutBs	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
FrTp	FrTp/FrTpMultipleConfig/FrTpConnectionControl
BSW Parameter	**BSW Type**
FrTpTimeoutCr	EcucFloatParamDef
BSW Description	
This parameter defines the timeout value in seconds a receiver is waiting for a CF or a LF.	
M2 Template	**M2 Description**
TPS_SYST	This parameter defines the timeout value in seconds for waiting for a CF or FF-x (in case of retry) after receiving the last CF or after sending an FC or AF on the receiver side. Specified in seconds.
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayTpConnectionControl.timeoutCr	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
FrTp	FrTp/FrTpMultipleConfig
BSW Parameter	**BSW Type**
FrTpMaxConnectionCnt	EcucIntegerParamDef
BSW Description	
Maximum number of TP connections. This parameter is needed only in case of post-build loadable implementation using static memory allocation.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context	
FrTp	FrTp/FrTpMultipleConfig	
BSW Parameter		**BSW Type**
FrTpRxPduPool		EcucParamConfContainerDef
BSW Description		
This container contains all Pdus that are assigned to that Pdu Pool.		
M2 Template	**M2 Description**	
TPS_SYST	FlexrayTpPduPool is a set of N-PDUs which are defined for FrTp sending or receiving purpose.	
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayTpPduPool		
Mapping Rule		**Mapping Type**
Create container if the FlexrayTpPduPool element is referenced by the Flexray TpConnection via the rxPduPool reference.		full

BSW Module	BSW Context	
FrTp	FrTp/FrTpMultipleConfig/FrTpRxPduPool	
BSW Parameter		**BSW Type**
FrTpRxPdu		EcucParamConfContainerDef
BSW Description		
Container to hold the PDU parameters. ImplementationType: PduInfoType		
M2 Template	**M2 Description**	
TPS_SYST	Reference to NPdus that are part of the PduPool.	
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayTpPduPool.nPdu		
Mapping Rule		**Mapping Type**
Create container for each NPdu that is referenced by the regarded FlexrayTp PduPool.		full

BSW Module	BSW Context	
FrTp	FrTp/FrTpMultipleConfig/FrTpRxPduPool/FrTpRxPdu	
BSW Parameter		**BSW Type**
FrTpRxPduId		EcucIntegerParamDef
BSW Description		
This is a unique identifier for a received message which is forwarded from the FrIf to the FrTp. ImplementationType: PduIdType		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrTp	FrTp/FrTpMultipleConfig/FrTpRxPduPool/FrTpRxPdu	
BSW Parameter		**BSW Type**
FrTpRxPduRef		EcucReferenceDef

BSW Description		
Reference to a PDU in the global PDU structure.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
FrTp	FrTp/FrTpMultipleConfig	
BSW Parameter		**BSW Type**
FrTpTxPduPool		EcucParamConfContainerDef
BSW Description		
This container contains all Pdus that are assigned to that Pdu Pool.		
M2 Template	**M2 Description**	
TPS_SYST	FlexrayTpPduPool is a set of N-PDUs which are defined for FrTp sending or receiving purpose.	
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayTpPduPool		
Mapping Rule		**Mapping Type**
Create container if the FlexrayTpPduPool element is referenced by the FlexrayTpConnection via the txPduPool reference.		full

BSW Module	**BSW Context**	
FrTp	FrTp/FrTpMultipleConfig/FrTpTxPduPool	
BSW Parameter		**BSW Type**
FrTpTxPdu		EcucParamConfContainerDef
BSW Description		
Container to hold the PDU parameters. ImplementationType: PduInfoType		
M2 Template	**M2 Description**	
TPS_SYST	Reference to NPdus that are part of the PduPool.	
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayTpPduPool.nPdu		
Mapping Rule		**Mapping Type**
Create container for each NPdu that is referenced by the regarded FlexrayTpPduPool.		full

BSW Module	**BSW Context**	
FrTp	FrTp/FrTpMultipleConfig/FrTpTxPduPool/FrTpTxPdu	
BSW Parameter		**BSW Type**
FrTpTxConfirmationPduId		EcucIntegerParamDef
BSW Description		
Handle Id to be used by the FrIf to confirm the transmission of the FrTpTxPdu to the FrIf module (FrTp_TxConfirmation) and for TriggerTransmit (FrTp_TriggerTransmit).		
M2 Template	**M2 Description**	
M2 Parameter		

Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**	
FrTp	FrTp/FrTpMultipleConfig/FrTpTxPduPool/FrTpTxPdu	
BSW Parameter		**BSW Type**
FrTpTxPduRef		EcucReferenceDef
BSW Description		
Reference to a PDU in the global PDU structure.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

D.4.5 FrArTp Mapping

BSW Module	**BSW Context**	
FrArTp	FrArTp	
BSW Parameter		**BSW Type**
FrArTpGeneral		EcucParamConfContainerDef
BSW Description		
This container contains the general configuration (parameters) of the FlexRay TP.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
FrArTp	FrArTp/FrArTpGeneral	
BSW Parameter		**BSW Type**
FrArTpChanNum		EcucIntegerParamDef
BSW Description		
This reference is deprecated and will be removed in a future release. Old description: Preprocessor switch for defining the number of concurrent channels the module supports. Up to 32 channels shall be definable here.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context		
FrArTp	FrArTp/FrArTpGeneral		
BSW Parameter		**BSW Type**	
FrArTpDevErrorDetect		EcucBooleanParamDef	
BSW Description			
Preprocessor switch for enabling development error detection.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
FrArTp	FrArTp/FrArTpGeneral		
BSW Parameter		**BSW Type**	
FrArTpHaveAckRt		EcucBooleanParamDef	
BSW Description			
Preprocessor switch for enabling the Acknowledgement and retry mechanisms.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
FrArTp	FrArTp/FrArTpGeneral		
BSW Parameter		**BSW Type**	
FrArTpHaveGrpSeg		EcucBooleanParamDef	
BSW Description			
Preprocessor switch for enabling segmentation of 1:n messages.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
FrArTp	FrArTp/FrArTpGeneral		
BSW Parameter		**BSW Type**	
FrArTpHaveLm		EcucBooleanParamDef	
BSW Description			
Preprocessor switch for enabling the mechanism for message longer than allowed by.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

	local

BSW Module	BSW Context	
FrArTp	FrArTp/FrArTpGeneral	
BSW Parameter		**BSW Type**
FrArTpHaveTc		EcucBooleanParamDef
BSW Description		
Preprocessor switch for enabling Transmit Cancellation and Receive Cancellation.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrArTp	FrArTp/FrArTpGeneral	
BSW Parameter		**BSW Type**
FrArTpMainFuncCycle		EcucFloatParamDef
BSW Description		
This parameter contains the calling period of the TPs Main Function. The parameter is specified in seconds.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrArTp	FrArTp/FrArTpGeneral	
BSW Parameter		**BSW Type**
FrArTpVersionInfoApi		EcucBooleanParamDef
BSW Description		
Preprocessor switch for enabling the Version info API.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrArTp	FrArTp	
BSW Parameter		**BSW Type**
FrArTpMultipleConfig		EcucParamConfContainerDef
BSW Description		
This container holds one or several multiple configuration sets.		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**	
FrArTp	FrArTp/FrArTpMultipleConfig	
BSW Parameter		**BSW Type**
FrArTpChannel		EcucParamConfContainerDef
BSW Description		
This container contains the configuration (parameters) of one FlexRay TP channel.		
M2 Template	**M2 Description**	
TPS_SYST	A channel is a group of connections sharing several properties. The FlexRayArTp supports several channels. These channels can work concurrently, thus each of them requires its own state machine and management data structures and its own PDU-IDs.	
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayArTpChannel		
Mapping Rule		**Mapping Type**
Create container for each FlexrayArTpChannel that exists in the Ecu Extract.		full

BSW Module	**BSW Context**	
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel	
BSW Parameter		**BSW Type**
FrArTpAckType		EcucEnumerationParamDef
BSW Description		
This parameter defines the type of acknowledgement which is used for the specific channel.		
M2 Template	**M2 Description**	
TPS_SYST	Type of Acknowledgement.	
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayArTpChannel.ackType		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel	
BSW Parameter		**BSW Type**
FrArTpAdrType		EcucEnumerationParamDef
BSW Description		
This parameter states the addressing type this connection has. The meanings of the values are one byte and two byte.		
M2 Template	**M2 Description**	
TPS_SYST	Adressing Type of this connection: true: Two Bytes false: One Byte	
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayArTpChannel.extendedAddressing		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context		
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel		
BSW Parameter		**BSW Type**	
FrArTpChannelId		EcucIntegerParamDef	
BSW Description			
Please note that this parameter is deprecated and will be removed in a future release. Old description: The Id of the channel.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel		
BSW Parameter		**BSW Type**	
FrArTpConNum		EcucIntegerParamDef	
BSW Description			
Please note that this parameter is deprecated and will be removed in a future release. Old description: This parameter states the number of connections used in this channel. At least 256 shall be configurable here.			
M2 Template	**M2 Description**		
TPS_SYST	Group of connections that can be used in this channel.		
M2 Parameter			
SystemTemplate::TransportProtocols::FlexrayArTpChannel.tpConnection			
Mapping Rule			**Mapping Type**
Count aggregated TpConnections.			full

BSW Module	BSW Context		
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel		
BSW Parameter		**BSW Type**	
FrArTpConcurrentConnections		EcucIntegerParamDef	
BSW Description			
This parameter defines the number of connections that can be active at the same time. If set to 0, all configured connections can be active at the same time.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context	
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel	
BSW Parameter		**BSW Type**

FrArTpConnection		EcucParamConfContainerDef	
BSW Description			
This container contains the configuration (parameters) of one FlexRay TP connection. A connection can only belong to one channel.			
M2 Template	**M2 Description**		
TPS_SYST	Group of connections that can be used in this channel.		
M2 Parameter			
SystemTemplate::TransportProtocols::FlexrayArTpChannel.tpConnection			
Mapping Rule			**Mapping Type**
Create container for each existing FlexrayArTpConnection that is aggregated by FlexrayArTpChannel in the System description.			full

BSW Module	**BSW Context**		
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel/FrArTpConnection		
BSW Parameter		**BSW Type**	
FrArTpConPduRef		EcucReferenceDef	
BSW Description			
This reference is deprecated and will be removed in a future release. It has been rendered irrelevant by the introduction of PDU pools. Old description: Each value defines a PDU to be used for this connection. Thus each value is a PDU-ID given in FrArTpPdu and this array cannot be longer than the array FrArTpPdu. Please note: Only PDUs of the same size shall be used within a connection. Of course the PDU having the TxConfirmation configured has to be used by every connection.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel/FrArTpConnection		
BSW Parameter		**BSW Type**	
FrArTpConPrioPdus		EcucIntegerParamDef	
BSW Description			
This parameter defines the number of TxNPdus to which this connection has prioritized access. It must be ensured that the number of prioritized PDUs of all connections is smaller than the total number of TxNPdus in the associated PDU pool.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**

FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel/FrArTpConnection		
BSW Parameter		**BSW Type**	
FrArTpLa		EcucIntegerParamDef	
BSW Description			
This parameter defines the Local Address for the respective connection. When the local instance is the sender, this is the Source Address within the TP frame. When the local instance is the receiver, this is the Target Address within the TP frame. Note that in case of 1 byte addressing only the values from 0x0000 - 0x00FF are valid.			
M2 Template	**M2 Description**		
TPS_SYST	The source of the TP connection.		
M2 Parameter			
SystemTemplate::TransportProtocols::FlexrayArTpConnection.source			
Mapping Rule			**Mapping Type**
LocalAddress can be derived from the TpNode that is referenced by the FlexRayTpConnection as source.			full

BSW Module	**BSW Context**		
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel/FrArTpConnection		
BSW Parameter		**BSW Type**	
FrArTpMultRec		EcucBooleanParamDef	
BSW Description			
This parameter defines, whether this connection is an 1:1 ('false') or an 1:n ('true') connection. Of course, if the channel to which the connection is configured has retry or acknowledgement enabled, no retry or acknowledgement will occur in case the connection is an 1:n connection.			
M2 Template	**M2 Description**		
TPS_SYST	TP address for 1:n connections.		
M2 Parameter			
SystemTemplate::TransportProtocols::FlexrayArTpConnection.multicast			
Mapping Rule			**Mapping Type**
If multicast is used set this attribute to true.			full

BSW Module	**BSW Context**		
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel/FrArTpConnection		
BSW Parameter		**BSW Type**	
FrArTpRa		EcucIntegerParamDef	
BSW Description			
This parameter defines the Remote Address for the respective connection. When the local instance is the sender, this is the Target Address within the TP frame. When the local instance is the receiver, this is the Source Address within the TP frame. Note that in case of 1 byte addressing only the values from 0x0000 - 0x00FF are valid.			
M2 Template	**M2 Description**		
TPS_SYST	The target of the TP connection.		
M2 Parameter			
SystemTemplate::TransportProtocols::FlexrayArTpConnection.target			
Mapping Rule			**Mapping Type**
RemoteAddress can be derived from the TpNode that is referenced by the FlexRayTpConnection as target.			full

BSW Module	**BSW Context**	
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel/FrArTpConnection	
BSW Parameter		**BSW Type**

FrArTpRxSdu		EcucParamConfContainerDef	
BSW Description			
Describes the Rx SDU			
M2 Template	**M2 Description**		
TPS_SYST	Reference to the IPdu that is segmented by the Transport Protocol.		
M2 Parameter			
SystemTemplate::TransportProtocols::FlexrayArTpConnection.directTpSdu			
Mapping Rule			**Mapping Type**
Create container for every IPdu that is received by the FrArTp and the regarded Ecu.			full

BSW Module	**BSW Context**		
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel/FrArTpConnection/FrArTpRxSdu		
BSW Parameter		**BSW Type**	
FrArTpRxSduRef		EcucReferenceDef	
BSW Description			
Reference to a PDU in the global PDU structure.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel/FrArTpConnection/FrArTpRxSdu		
BSW Parameter		**BSW Type**	
FrArTpSduRxId		EcucIntegerParamDef	
BSW Description			
This is a unique identifier for a received message. This Id is used in the CancelReceive and ChangeParameter API call. ImplementationType: PduIdType			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**	
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel/FrArTpConnection	
BSW Parameter		**BSW Type**
FrArTpTxSdu		EcucParamConfContainerDef
BSW Description		
Describes the Tx SDU		
M2 Template	**M2 Description**	
TPS_SYST	Reference to the IPdu that is segmented by the Transport Protocol.	
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayArTpConnection.directTpSdu		

Mapping Rule	Mapping Type
Create container for every IPdu that is transmitted by the FrArTp and the regarded Ecu.	full

BSW Module	BSW Context	
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel/FrArTpConnection/FrArTpTxSdu	
BSW Parameter		**BSW Type**
FrArTpSduTxId		EcucIntegerParamDef
BSW Description		
This is a unique identifier for a received or a to be transmitted message. With this (and by means of e.g. a lookup table) the PDU Router can route the message appropriately without dealing with the particularities of the Transport Layer. This parameter can also be seen as the identifier of a connection. ImplementationType: PduIdType		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel/FrArTpConnection/FrArTpTxSdu	
BSW Parameter		**BSW Type**
FrArTpTxSduRef		EcucReferenceDef
BSW Description		
Reference to a PDU in the global PDU structure.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel	
BSW Parameter		**BSW Type**
FrArTpGrpSeg		EcucBooleanParamDef
BSW Description		
Here can be specified, whether segmentation within a 1:n connection is allowed or not.		
M2 Template	**M2 Description**	
TPS_SYST	This attribute defines whether segmentation within a 1:n connection is allowed or not.	
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayArTpChannel.multicastSegmentation		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel	
BSW Parameter		**BSW Type**
FrArTpLm		EcucEnumerationParamDef
BSW Description		
This specifies the maximum message length for the particular channel.		
M2 Template	**M2 Description**	
TPS_SYST	This specifies the maximum message length for the particular channel.	
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayArTpChannel.maximumMessageLength		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel	
BSW Parameter		**BSW Type**
FrArTpMaxAr		EcucIntegerParamDef
BSW Description		
This parameter defines the maximum number of trying to send a frame when a TIMEOUT AR occurs.		
M2 Template	**M2 Description**	
TPS_SYST	This attribute defines the maximum number of trying to send a frame when a TIMEOUT AR occurs (depending on whether retry is configured).	
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayArTpChannel.maxAr		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel	
BSW Parameter		**BSW Type**
FrArTpMaxAs		EcucIntegerParamDef
BSW Description		
This parameter defines the maximum number of trying to send a frame when a TIMEOUT AS occurs.		
M2 Template	**M2 Description**	
TPS_SYST	This attribute defines the maximum number of trying to send a frame when a TIMEOUT AS occurs (depending on whether retry is configured).	
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayArTpChannel.maxAs		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel	
BSW Parameter		**BSW Type**
FrArTpMaxBs		EcucIntegerParamDef
BSW Description		
This parameter defines the number of consecutive CFs between two FCs (block size). Valid values are 1 .. 16 when retry is activated, and 0 .. 255 otherwise.		
M2 Template	**M2 Description**	

TPS_SYST	This attribute defines the number of consecutive CFs between two FCs (block size). Valid values are 1 .. 16 when retry is activated, and 0 .. 255 otherwise.
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayArTpChannel.maxBs	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel
BSW Parameter	**BSW Type**
FrArTpMaxBufReq	EcucIntegerParamDef
BSW Description	
Please note that this parameter is deprecated and will be removed in a future release. FrArTpMaxWft will be used instead to configure the maximum number of wait frames on receiver side. On the sender side, timeCs defines the maximum time for retries. Old description: This parameter defines the maximum number of times the FrArTp should send a wait frame FC(WT). It is also used to limit the number of retries for PduR_FrArTpCopyTxData and PduR_FrArTpCopyRxData when no timer is active.	
M2 Template	**M2 Description**
	Attribute is deprecated
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel
BSW Parameter	**BSW Type**
FrArTpMaxFrIf	EcucIntegerParamDef
BSW Description	
Please note that this parameter is deprecated and will be removed in the future. Old description: This parameter defines the maximum number of trying to send a frame when the FrIf returns an error.	
M2 Template	**M2 Description**
TPS_SYST	This attribute defines the maximum number of trying to send a frame when the FrIf returns an error.
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayArTpChannel.maxFrIf	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel
BSW Parameter	**BSW Type**
FrArTpMaxRn	EcucIntegerParamDef
BSW Description	

This parameter defines the maximum number of retries (if retry is configured for the particular channel).		
M2 Template	**M2 Description**	
TPS_SYST	This attribute defines the maximum number of retries (if retry is configured for the particular channel).	
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayArTpChannel.maxRetries		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel	
BSW Parameter		**BSW Type**
FrArTpMaxWft		EcucIntegerParamDef
BSW Description		
This parameter defines the maximal number of wait frames to be sent for a pending connection.		
M2 Template	**M2 Description**	
TPS_SYST	This attribute defines the maximal number of wait frames to be sent for a pending connection.	
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayArTpChannel.maxFcWait		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel	
BSW Parameter		**BSW Type**
FrArTpPdu		EcucParamConfContainerDef
BSW Description		
Container to hold the PDU parameters. ImplementationType: PduInfoType		
M2 Template	**M2 Description**	
TPS_SYST	A FlexRayTpChannel references a set of NPdus. These NPdus are logically assembled into a pool of Rx NPdus and another pool of Tx NPdus.	
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayArTpChannel.nPdu		
Mapping Rule		**Mapping Type**
Create container if NPdus are referenced by the FlexrayArTpChannel.		full

BSW Module	**BSW Context**	
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel/FrArTpPdu	
BSW Parameter		**BSW Type**
FrArTpPduDirection		EcucEnumerationParamDef
BSW Description		
This parameter defines the direction of the PDU.		
M2 Template	**M2 Description**	
TPS_SYST	Communication direction of the Connector Port (input or output Port).	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::PduTriggering.iPduPort		

Mapping Rule	Mapping Type
The direction of the Npdu can be derived from the triggering elements that contain references to IN- and OUT-Ports.	full

BSW Module	BSW Context	
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel/FrArTpPdu	
BSW Parameter		**BSW Type**
FrArTpPduId		EcucIntegerParamDef
BSW Description		
This is the identifier of the FlexRay Interface PDUs (Fr N-PDU, Fr L-SDU) in which the Transport Layer Frames of this channel should be transmitted. For FrArTpPduDirection == FRARTP_RX, this parameter specifies the ID that is used by FrIf when calling FrArTp_RxIndication, while for FrArTpPduDirection == FRARTP_TX this ID is used by FrIf when calling FrArTp_TxConfirmation or FrArTp_TriggerTransmit. ImplementationType: PduIdType		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel/FrArTpPdu	
BSW Parameter		**BSW Type**
FrArTpPduRef		EcucReferenceDef
BSW Description		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel	
BSW Parameter		**BSW Type**
FrArTpPduFc		EcucParamConfContainerDef
BSW Description		
Please note that this container with all references and parameters is deprecated and will be removed in the next major version. It has been rendered irrelevant by the introduction of PDU pools. Old description: This is the identifier of the FlexRay Interface PDUs (Fr N-PDU, Fr L-SDU) in which the Transport Layer Flow Control and Acknowledgement Frames of this channel should be transmitted. ImplementationType: PduInfoType		
M2 Template	**M2 Description**	

TPS_SYST	A FlexRayTpChannel references a set of NPdus. These NPdus are logically assembled into a pool of Rx NPdus and another pool of Tx NPdus.	
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayArTpChannel.nPdu		
Mapping Rule		**Mapping Type**
		full

BSW Module	**BSW Context**	
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel/FrArTpPduFc	
BSW Parameter		**BSW Type**
FrArTpPduFcDirection		EcucEnumerationParamDef
BSW Description		
This parameter defines the direction of the PDU. This parameter is deprecated and will be removed in the next major version.		
M2 Template	**M2 Description**	
TPS_SYST	Communication direction of the Connector Port (input or output Port).	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::PduTriggering.iPduPort		
Mapping Rule		**Mapping Type**
The direction of the Npdu can be derived from the triggering elements that contain references to IN- and OUT-Ports.		full

BSW Module	**BSW Context**	
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel/FrArTpPduFc	
BSW Parameter		**BSW Type**
FrArTpPduFcId		EcucIntegerParamDef
BSW Description		
This is the identifier of the FlexRay Interface PDUs (Fr N-PDU, Fr L-SDU) in which the Transport Layer Flow Control and Acknowledgement Frames of this channel should be transmitted. This parameter is deprecated and will be removed in the next major version.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel/FrArTpPduFc	
BSW Parameter		**BSW Type**
FrArTpPduFcRef		EcucReferenceDef
BSW Description		
This reference is deprecated and will be removed in the next major version.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel	
BSW Parameter		**BSW Type**
FrArTpStMin		EcucFloatParamDef
BSW Description		
This parameter defines the minimum amount of time between two succeeding CFs of a 1:1 segmented transmission in seconds. Valid values are 0, 100μs, 200μs .. 900μs, 1ms, 2ms .. 127ms. The value can be changed at runtime using the FrArTp_ChangeParameter interface. FrArTpStMin must be an integer multiple of the cycle length multiplied with the multiplexing factor, i.e. FrArTpStMin = n * FrIfGdCycle * m, where n is an integer >= 0 and m is the cycle multiplexor of those cycles where PDUs of the PDU pool are scheduled. Please note: Due to the scheduling strategies of FrArTp, FrArTpStMin can only be kept to a degree defined by the maximum temporal distance of the PDUs of a PDU pool within one FlexRay cycle.		
M2 Template	**M2 Description**	
TPS_SYST	This attribute defines the minimum amount of time between two succeeding CFs of a 1:1 segmented transmission in seconds. Valid values are 0, 100μs, 200μs .. 900μs, 1ms, 2ms .. 127ms.	
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayArTpChannel.minimumSeparationTime		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel	
BSW Parameter		**BSW Type**
FrArTpStMinGrpSeg		EcucFloatParamDef
BSW Description		
This parameter defines the minimum amount of time between two succeeding CFs of a 1:n segmented transmission in seconds. Valid values are 0, 100μs, 200μs ... 900μs, 1ms, 2ms .. 127ms. The value can be changed at runtime using the FrArTp_ChangeParameter interface. FrArTpStMinGrpSeg must be an integer multiple of the cycle length multiplied with the multiplexing factor, i.e. FrArTpStMinGrpSeg = n * FrIfGdCycle * m, where n is an integer >= 0 and m is the cycle multiplexor of those cycles where PDUs of the PDU pool are scheduled. Please note: Due to the scheduling strategies of FrArTp, FrArTpStMinGrpSeg can only be kept to a degree defined by the maximum temporal distance of the PDUs of a PDU pool within one FlexRay cycle.		
M2 Template	**M2 Description**	
TPS_SYST	This attribute defines the minimum amount of time between two succeeding CFs of a 1:n segmented transmission in seconds.	
M2 Parameter		
SystemTemplate::TransportProtocols::FlexrayArTpChannel.minimumMulticastSeperationTime		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel	
BSW Parameter		**BSW Type**

FrArTpTc	EcucBooleanParamDef
BSW Description	
With this switch Transmit Cancellation and Receive Cancellation can be turned on or off for this channel.	
M2 Template	**M2 Description**
TPS_SYST	With this switch Tx and Rx Cancellation can be turned on or off.
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayArTpChannel.cancellation	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel
BSW Parameter	**BSW Type**
FrArTpTimeBr	EcucFloatParamDef
BSW Description	
This parameter defines the time in seconds between receiving the last CF of a block or an FF-x (or SF-x) and sending out an FC or AF. It is obvious that FRARTP_TIME_BR + (FRARTP_TIMEOUT_AR * FRARTP_MAX_AR) < FRARTP_TIMEOUT_BS must hold (because the transmission duration on the bus has also to be considered). This parameter is defined in ISO 15765-2. It is contained in the configuration as a performance requirement.	
M2 Template	**M2 Description**
TPS_SYST	This attribute defines the time in seconds between receiving the last CF of a block or an FF-x (or SF-x) and sending out an FC or AF.
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayArTpChannel.timeBr	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel
BSW Parameter	**BSW Type**
FrArTpTimeBuffer	EcucFloatParamDef
BSW Description	
Please note that this parameter is deprecated and will be removed in a future release. FrArTpTimeBr will be used instead to configure the delay between two wait frames (and thus two buffer requests) on receiver side. On sender side, the main task cycle will be used. Old description: This parameter defines the time in seconds of waiting for the next try to get a Tx or Rx buffer.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

<table>
<tr><th>BSW Module</th><th colspan="2">BSW Context</th></tr>
<tr><td>FrArTp</td><td colspan="2">FrArTp/FrArTpMultipleConfig/FrArTpChannel</td></tr>
<tr><th colspan="2">BSW Parameter</th><th>BSW Type</th></tr>
<tr><td colspan="2">FrArTpTimeCs</td><td>EcucFloatParamDef</td></tr>
<tr><th colspan="3">BSW Description</th></tr>
<tr><td colspan="3">This parameter defines the time in seconds between the sending of two consecutive CFs or between reception of an FC or AF and sending of the next CF .

It is obvious that FRARTP_TIME_CS + (FRARTP_TIMEOUT_AS * FRARTP_MAX_AS) $<$ FRARTP_TIMEOUT_CR must hold (because the transmission duration on the bus has also to be considered).

This parameter is defined in ISO 15765-2. It is contained in the configuration as a performance requirement.</td></tr>
<tr><th>M2 Template</th><th colspan="2">M2 Description</th></tr>
<tr><td>TPS_SYST</td><td colspan="2">Defines the time between the sending of two consecutive frames or between a consecutive frame and a flow control or between reception of an flow control or Acknowledgement Frame and sending of the next consecutive frame or a flow control.</td></tr>
<tr><th colspan="3">M2 Parameter</th></tr>
<tr><td colspan="3">SystemTemplate::TransportProtocols::FlexrayArTpChannel.timeCs</td></tr>
<tr><th colspan="2">Mapping Rule</th><th>Mapping Type</th></tr>
<tr><td colspan="2">1:1 mapping</td><td>full</td></tr>
</table>

<table>
<tr><th>BSW Module</th><th colspan="2">BSW Context</th></tr>
<tr><td>FrArTp</td><td colspan="2">FrArTp/FrArTpMultipleConfig/FrArTpChannel</td></tr>
<tr><th colspan="2">BSW Parameter</th><th>BSW Type</th></tr>
<tr><td colspan="2">FrArTpTimeFrIf</td><td>EcucFloatParamDef</td></tr>
<tr><th colspan="3">BSW Description</th></tr>
<tr><td colspan="3">Please note that this parameter is deprecated and will be removed in the future.

Old description:
This parameter defines the time in seconds of waiting for the next try to send via FrIf_Transmit.</td></tr>
<tr><th>M2 Template</th><th colspan="2">M2 Description</th></tr>
<tr><td>TPS_SYST</td><td colspan="2">This attribute defines the time in seconds of waiting for the next try (if retry is activated) to send via FrIf_Transmit. Specified in seconds.</td></tr>
<tr><th colspan="3">M2 Parameter</th></tr>
<tr><td colspan="3">SystemTemplate::TransportProtocols::FlexrayArTpChannel.timeFrIf</td></tr>
<tr><th colspan="2">Mapping Rule</th><th>Mapping Type</th></tr>
<tr><td colspan="2">1:1 mapping</td><td>full</td></tr>
</table>

<table>
<tr><th>BSW Module</th><th colspan="2">BSW Context</th></tr>
<tr><td>FrArTp</td><td colspan="2">FrArTp/FrArTpMultipleConfig/FrArTpChannel</td></tr>
<tr><th colspan="2">BSW Parameter</th><th>BSW Type</th></tr>
<tr><td colspan="2">FrArTpTimeoutAr</td><td>EcucFloatParamDef</td></tr>
<tr><th colspan="3">BSW Description</th></tr>
<tr><td colspan="3">This parameter states the timeout in seconds between the PDU transmit request of the Transport Layer to the FlexRay Interface and the corresponding confirmation of the FlexRay Interface on the receiver side (for FC or AF).</td></tr>
<tr><th>M2 Template</th><th colspan="2">M2 Description</th></tr>
</table>

TPS_SYST	This attribute states the timeout in seconds between the PDU transmit request of the Transport Layer to the FlexRay Interface and the corresponding confirmation of the FlexRay Interface on the receiver side (for FC or AF).
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayArTpChannel.timeoutAr	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel
BSW Parameter	**BSW Type**
FrArTpTimeoutAs	EcucFloatParamDef
BSW Description	
This parameter states the timeout in seconds between the PDU transmit request for the first PDU of the group used in the current connection of the Transport Layer to the FlexRay Interface and the corresponding confirmation of the FlexRay Interface (when having sent the last PDU of the group used in this connection) on the sender side (SF-x, FF-x, CF).	
M2 Template	**M2 Description**
TPS_SYST	This attribute states the timeout in seconds between the PDU transmit request for the first PDU of the group used in the current connection of the Transport Layer to the FlexRay Interface and the corresponding confirmation of the FlexRay Interface.
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayArTpChannel.timeoutAs	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel
BSW Parameter	**BSW Type**
FrArTpTimeoutBs	EcucFloatParamDef
BSW Description	
This parameter defines the timeout in seconds for waiting for an FC or AF on the sender side in a 1:1 connection.	
M2 Template	**M2 Description**
TPS_SYST	This attribute defines the timeout in seconds for waiting for an FC or AF on the sender side in a 1:1 connection.
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayArTpChannel.timeoutBs	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel
BSW Parameter	**BSW Type**
FrArTpTimeoutCr	EcucFloatParamDef
BSW Description	
This parameter defines the timeout value in seconds for waiting for a CF or FF-x (in case of retry) after receiving the last CF or after sending an FC or AF on the receiver side.	
M2 Template	**M2 Description**

TPS_SYST	This attribute defines the timeout value in seconds for waiting for a CF or FF-x (in case of retry) after receiving the last CF or after sending an FC or AF on the receiver side.
M2 Parameter	
SystemTemplate::TransportProtocols::FlexrayArTpChannel.timeoutCr	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
FrArTp	FrArTp/FrArTpMultipleConfig/FrArTpChannel
BSW Parameter	**BSW Type**
FrArTpUsePduFc	EcucBooleanParamDef
BSW Description	
Please note that this parameter is deprecated and will be removed in a future release. It has been rendered irrelevant by the introduction of PDU pools. Old description: This switch defines, whether within this channel the dedicated FC/ACK PDU (FrArTpPduFc) shall be used or not. If this is not used FC / ACK frames are sent using the normal IDs, otherwise only FrArTpPduFc shall be used for sending / receiving FC / ACK frames.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

D.4.6 FrSM Mapping

BSW Module	**BSW Context**
FrSM	FrSM
BSW Parameter	**BSW Type**
FrSMConfig	EcucParamConfContainerDef
BSW Description	
This container comprises the cluster specific configuration of the FlexRay State Manager.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**
FrSM	FrSM/FrSMConfig
BSW Parameter	**BSW Type**
FrSMCluster	EcucParamConfContainerDef
BSW Description	
This container specifies a FlexRay cluster and all related data. A FlexRay cluster may consist of more than one controller per ECU.	

M2 Template	M2 Description	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
FrSM	FrSM/FrSMConfig/FrSMCluster	
BSW Parameter		**BSW Type**
FrSMCheckWakeupReason		EcucBooleanParamDef
BSW Description		
If FrSMCheckWakeupReason is true, the FrSM will check the wakeup reason in order to skip the wakeup in case of wakeup by bus. If FrSMCheckWakeupReason is false, the FrSM will always try to perform a wakeup.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
FrSM	FrSM/FrSMConfig/FrSMCluster	
BSW Parameter		**BSW Type**
FrSMClusterDemEventParameterRefs		EcucParamConfContainerDef
BSW Description		
Container for the references to DemEventParameter elements which shall be invoked using the API Dem_ReportErrorStatus in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId value. The standardized errors are provided in this container and can be extended by vendor specific error references.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
FrSM	FrSM/FrSMConfig/FrSMCluster/FrSMClusterDemEventParameterRefs	
BSW Parameter		**BSW Type**
FRSM_E_CLUSTER_STARTUP		EcucSymbolicNameReferenceDef
BSW Description		
Reference to the DemEventParameter which shall be issued when the error "FRSM_E_CLUSTER_STARTUP" has occurred. If the reference is not configured the error shall be reported as DET error.		
M2 Template	**M2 Description**	
M2 Parameter		

Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**	
FrSM	FrSM/FrSMConfig/FrSMCluster/FrSMClusterDemEventParameterRefs	
BSW Parameter		**BSW Type**
FRSM_E_CLUSTER_SYNC_LOSS		EcucSymbolicNameReferenceDef
BSW Description		
Reference to the DemEventParameter which shall be issued when the error "FRSM_E_CLUSTER_SYNC_LOSS" has occurred. If the reference is not configured the error shall be reported as DET error.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
FrSM	FrSM/FrSMConfig/FrSMCluster	
BSW Parameter		**BSW Type**
FrSMComMNetworkHandleRef		EcucSymbolicNameReferenceDef
BSW Description		
Reference to the unique handle to identify one certain FlexRay network correspond to one of the network handles of the ComM configuration.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
FrSM	FrSM/FrSMConfig/FrSMCluster	
BSW Parameter		**BSW Type**
FrSMDelayStartupWithoutWakeup		EcucBooleanParamDef
BSW Description		
If true, timer t1 shall be started instead of immediately calling FrIf_AllowColdstart in case of a startup without wakeup.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**
FrSM	FrSM/FrSMConfig/FrSMCluster

BSW Parameter		BSW Type	
FrSMDurationT1		EcucFloatParamDef	
BSW Description			
The duration of timer t1 in seconds. A value of 0 shall imply that the timer is not used.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	BSW Context		
FrSM	FrSM/FrSMConfig/FrSMCluster		
BSW Parameter		**BSW Type**	
FrSMDurationT2		EcucFloatParamDef	
BSW Description			
The duration of timer t2 in seconds. A value of 0 shall imply that the timer is not used. The value of this parameter shall be larger than the value of FrSMDurationT1 parameter.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	BSW Context		
FrSM	FrSM/FrSMConfig/FrSMCluster		
BSW Parameter		**BSW Type**	
FrSMDurationT3		EcucFloatParamDef	
BSW Description			
The duration of timer t3 in seconds. The value of this parameter shall be larger than the value of FrSMDurationT1 parameter. A value of 0 shall imply that the timer is not used. It shall only be possible to configure a value 0 if no FrNm is used.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	BSW Context	
FrSM	FrSM/FrSMConfig/FrSMCluster	
BSW Parameter		**BSW Type**
FrSMFrIfClusterRef		EcucSymbolicNameReferenceDef
BSW Description		
References the cluster configuration in the FlexRay Interface configuration. Note that the assigned controllers and transceivers are defined in the FrIf configuration and can be accessed via this reference.		

M2 Template	M2 Description	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
FrSM	FrSM/FrSMConfig/FrSMCluster	
BSW Parameter		**BSW Type**
FrSMIsColdstartEcu		EcucBooleanParamDef
BSW Description		
True: The ECU is a coldstart node for this FlexRay cluster. False: The ECU is no coldstart node for this FlexRay cluster.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
FrSM	FrSM/FrSMConfig/FrSMCluster	
BSW Parameter		**BSW Type**
FrSMIsWakeupEcu		EcucBooleanParamDef
BSW Description		
True: FrSM shall perform a wakeup for this cluster. False: FrSM shall never perform a wakeup for this FlexRay cluster.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
FrSM	FrSM/FrSMConfig/FrSMCluster	
BSW Parameter		**BSW Type**
FrSMMainFunctionCycleTime		EcucFloatParamDef
BSW Description		
This parameter defines the cycle time in seconds of the periodic calling of FrSM main function.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context

FrSM	FrSM/FrSMConfig/FrSMCluster	
BSW Parameter		**BSW Type**
FrSMMinNumberOfColdstarter		EcucIntegerParamDef
BSW Description		
This parameter defines the number of coldstarter that should not be underrun. If this parameter is not configured the mainfunction shall not check the number of startup frames.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
FrSM	FrSM/FrSMConfig/FrSMCluster	
BSW Parameter		**BSW Type**
FrSMNumWakeupPatterns		EcucIntegerParamDef
BSW Description		
Maximum number of Wakeup Patterns the node may send before going to FRSM_STARTUP.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
FrSM	FrSM/FrSMConfig/FrSMCluster	
BSW Parameter		**BSW Type**
FrSMStartupRepetitions		EcucIntegerParamDef
BSW Description		
The number of times an ECU may repeat the startup procedure for a FlexRay cluster.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
FrSM	FrSM/FrSMConfig/FrSMCluster	
BSW Parameter		**BSW Type**
FrSMStartupRepetitionsWithWakeup		EcucIntegerParamDef
BSW Description		
The number of times an ECU may repeat the startup procedure including a wakeup for a FlexRay cluster.		
M2 Template	**M2 Description**	
M2 Parameter		

Mapping Rule	Mapping Type

BSW Module	BSW Context	
FrSM	FrSM/FrSMConfig/FrSMCluster	
BSW Parameter		**BSW Type**
FrSMTrcvStdbyDelay		EcucFloatParamDef
BSW Description		
The duration of timer t_TrcvStdbyDelay in seconds. The granularity of this parameter shall be restricted to full FlexRay cycles (FrIfGdCycle). A value of 0 shall imply that the timer is not used.		
M2 Template	**M2 Description**	
TPS_SYST	The duration of timer t_TrcvStdbyDelay in seconds. The granularity of this parameter shall be restricted to full FlexRay cycles (cycle). The transceiver status setting to STANDBY shall be delayed by this value.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology::FlexrayCluster.tranceiverStandbyDelay		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
FrSM	FrSM	
BSW Parameter		**BSW Type**
FrSMGeneral		EcucParamConfContainerDef
BSW Description		
This container contains the general configuration parameters of the FlexRay State Manager.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
FrSM	FrSM/FrSMGeneral	
BSW Parameter		**BSW Type**
FrSMAllSlotsSupport		EcucBooleanParamDef
BSW Description		
Configuration parameter to enable/disable FrSM support to enable/disable the switching from key-slot/single-slot mode to all-slot mode.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context

FrSM	FrSM/FrSMGeneral
BSW Parameter	**BSW Type**
FrSMCddHeaderFile	EcucStringParamDef
BSW Description	
This parameter defines header files for callback functions which are implemented by CDD, e.g. <Cdd>_SyncLossErrorIndication.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**
FrSM	FrSM/FrSMGeneral
BSW Parameter	**BSW Type**
FrSMDevErrorDetect	EcucBooleanParamDef
BSW Description	
Enables and disables the development error detection and notification mechanism.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**
FrSM	FrSM/FrSMGeneral
BSW Parameter	**BSW Type**
FrSMSyncLossErrorIndicationName	EcucFunctionNameDef
BSW Description	
Name of <Cdd>_SyncLossErrorIndication function that shall be called on loss of synchronization. If this parameter is omitted no indication shall take place.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**
FrSM	FrSM/FrSMGeneral
BSW Parameter	**BSW Type**
FrSMVersionInfoApi	EcucBooleanParamDef
BSW Description	
Enables and disables the version info API	
M2 Template	**M2 Description**
M2 Parameter	

Mapping Rule	**Mapping Type**

D.5 Lin

D.5.1 Lin Driver Mapping

BSW Module	**BSW Context**	
Lin	Lin	
BSW Parameter		**BSW Type**
LinDemEventParameterRefs		EcucParamConfContainerDef
BSW Description		
Container for the references to DemEventParameter elements which shall be invoked using the API Dem_ReportErrorStatus API in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId value. The standardized errors are provided in the container and can be extended by vendor specific error references.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
Lin	Lin/LinDemEventParameterRefs	
BSW Parameter		**BSW Type**
LIN_E_TIMEOUT		EcucSymbolicNameReferenceDef
BSW Description		
Reference to the DemEventParameter which shall be issued when the error "Timeout caused by hardware error" has occurred. If the reference is not configured the error shall be reported as DET error.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
Lin	Lin	
BSW Parameter		**BSW Type**
LinGeneral		EcucParamConfContainerDef
BSW Description		
This container contains the parameters related to each LIN Driver Unit.		
M2 Template	**M2 Description**	
M2 Parameter		

Mapping Rule	Mapping Type
	local

BSW Module	BSW Context	
Lin	Lin/LinGeneral	
BSW Parameter		**BSW Type**
LinDevErrorDetect		EcucBooleanParamDef
BSW Description		
Switches the Development Error Detection and Notification ON or OFF.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Lin	Lin/LinGeneral	
BSW Parameter		**BSW Type**
LinIndex		EcucIntegerParamDef
BSW Description		
Specifies the InstanceId of this module instance. If only one instance is present it shall have the Id 0.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Lin	Lin/LinGeneral	
BSW Parameter		**BSW Type**
LinTimeoutDuration		EcucIntegerParamDef
BSW Description		
Specifies the maximum number of loops for blocking function until a timeout is raised in short term wait loops		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Lin	Lin/LinGeneral	
BSW Parameter		**BSW Type**
LinVersionInfoApi		EcucBooleanParamDef
BSW Description		

Switches the Lin_GetVersionInfo function ON or OFF.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
Lin	Lin	
BSW Parameter		**BSW Type**
LinGlobalConfig		EcucParamConfContainerDef
BSW Description		
This container contains the global configuration parameter of the Lin driver. This container is a MultipleConfigurationContainer, i.e. this container and its sub-containers exit once per configuration set.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
Lin	Lin/LinGlobalConfig	
BSW Parameter		**BSW Type**
LinChannel		EcucParamConfContainerDef
BSW Description		
This container contains the configuration (parameters) of the LIN Controller(s).		
M2 Template	**M2 Description**	
TPS_SYST	A physical channel is the transmission medium that is used to send and receive information between two communicating ECUs. Each CommunicationCluster has at least one physical channel.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreTopology::PhysicalChannel		
Mapping Rule		**Mapping Type**
A LinChannel container is constructed per CommunicationConnector belonging to the CommunicationController associated with the owning Lin Module container		full

BSW Module	**BSW Context**	
Lin	Lin/LinGlobalConfig/LinChannel	
BSW Parameter		**BSW Type**
LinChannelBaudRate		EcucIntegerParamDef
BSW Description		
Specifies the baud rate of the LIN channel		
M2 Template	**M2 Description**	
TPS_SYST	Channels speed in bits/s.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreTopology::CommunicationCluster.baudrate		

Mapping Rule	Mapping Type
1:1 mapping	full

BSW Module	BSW Context	
Lin	Lin/LinGlobalConfig/LinChannel	
BSW Parameter		**BSW Type**
LinChannelEcuMWakeupSource		EcucSymbolicNameReferenceDef
BSW Description		
This parameter contains a reference to the Wakeup Source for this controller as defined in the ECU State Manager.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Lin	Lin/LinGlobalConfig/LinChannel	
BSW Parameter		**BSW Type**
LinChannelId		EcucIntegerParamDef
BSW Description		
Identifies the LIN channel. Replaces LIN_CHANNEL_INDEX_NAME from the LIN SWS.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
Implicit from each CommunicationConnector on the ECU representing a LIN channel. Increase the LinChannelId for each LIN channel created on the same CommunicationController, for each CommunicationController start indexing at zero.		local

BSW Module	BSW Context	
Lin	Lin/LinGlobalConfig/LinChannel	
BSW Parameter		**BSW Type**
LinChannelWakeupSupport		EcucBooleanParamDef
BSW Description		
Specifies if the LIN hardware channel supports wake up functionality		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
Lin	Lin/LinGlobalConfig/LinChannel	
BSW Parameter		**BSW Type**

LinClockRef		EcucReferenceDef	
BSW Description			
Reference to the LIN clock source configuration, which is set in the MCU driver configuration.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

D.5.2 Lin Interface Mapping

BSW Module	**BSW Context**		
LinIf	LinIf		
BSW Parameter		**BSW Type**	
LinIfGeneral		EcucParamConfContainerDef	
BSW Description			
This container contains the general parameters of LIN Interface module.			
M2 Template	**M2 Description**		
TPS_SYST	The CommunicationCluster is the main element to describe the topological connection of communicating ECUs.		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinCluster			
Mapping Rule			**Mapping Type**
Container must be created if the ECU is connected to a LIN Cluster			full

BSW Module	**BSW Context**		
LinIf	LinIf/LinIfGeneral		
BSW Parameter		**BSW Type**	
LinIfCancelTransmitSupported		EcucBooleanParamDef	
BSW Description			
Global Pre-Compile Switch to enable/disable the APIs LinIf_CancelTransmit/LinTp_CancelReceive.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
LinIf	LinIf/LinIfGeneral		
BSW Parameter		**BSW Type**	
LinIfDevErrorDetect		EcucBooleanParamDef	
BSW Description			
Switches the Development Error Detection and Notification ON or OFF.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

	local

BSW Module	BSW Context	
LinIf	LinIf/LinIfGeneral	
BSW Parameter		**BSW Type**
LinIfMultipleDriversSupported		EcucBooleanParamDef
BSW Description		
States if multiple drivers are supported by the LIN Interface or not. The reason for this parameter is to reduce the size of LIN Interface if multiple drivers are not used.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
LinIf	LinIf/LinIfGeneral	
BSW Parameter		**BSW Type**
LinIfMultipleTrcvDriverSupported		EcucBooleanParamDef
BSW Description		
States if multiple transceiver drivers are supported by the LIN Interface or not. The reason for this parameter is to reduce the size of LIN Interface if multiple transceiver drivers are not used.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
LinIf	LinIf/LinIfGeneral	
BSW Parameter		**BSW Type**
LinIfNcOptionalRequestSupported		EcucBooleanParamDef
BSW Description		
States if the node configuration commands Assign NAD and Conditional Change NAD are supported.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
LinIf	LinIf/LinIfGeneral	
BSW Parameter		**BSW Type**
LinIfPublicCddHeaderFile		EcucStringParamDef
BSW Description		

Defines header files for callback functions which shall be included in case of CDDs. Range of characters is 1.. 32.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**	
LinIf	LinIf/LinIfGeneral	
BSW Parameter		**BSW Type**
LinIfTpSupported		EcucBooleanParamDef
BSW Description		
States if the TP is included in the LIN Interface or not. The reason for this parameter is to reduce the size of LIN Interface if the TP is not used.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
LinIf	LinIf/LinIfGeneral	
BSW Parameter		**BSW Type**
LinIfTrcvDriverSupported		EcucBooleanParamDef
BSW Description		
States if transceiver driver support is included in the LIN Interface or not. The reason for this parameter is to reduce the size of LIN Interface if transceiver drivers are not used.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
LinIf	LinIf/LinIfGeneral	
BSW Parameter		**BSW Type**
LinIfVersionInfoApi		EcucBooleanParamDef
BSW Description		
Switches the LinIf_GetVersionInfo function ON or OFF.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
LinIf	LinIf	
BSW Parameter		**BSW Type**
LinIfGlobalConfig		EcucParamConfContainerDef
BSW Description		
This container contains the global configuration parameters of the LinIf. It is a MultipleConfigurationContainer, i.e. this container and its sub-containers exist once per configuration set.		
M2 Template	**M2 Description**	
TPS_SYST	The CommunicationCluster is the main element to describe the topological connection of communicating ECUs.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinCluster		
Mapping Rule		**Mapping Type**
Container must be created if the ECU is connected to a LIN Cluster		full

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig	
BSW Parameter		**BSW Type**
LinIfChannel		EcucParamConfContainerDef
BSW Description		
Describes each LIN channel the LinIf is connected to.		
M2 Template	**M2 Description**	
TPS_SYST	The connection between the referencing ECU and the referenced channel via the referenced controller.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreTopology::CommunicationConnector		
Mapping Rule		**Mapping Type**
Container must be created if the CommunicationConnector belonging to the EC U is connected to a LinChannel.		full

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel	
BSW Parameter		**BSW Type**
LinIfCddRef		EcucForeignReferenceDef
BSW Description		
Reference to the CDD module description. This parameter is only required when LinIfWakeupConfirmationUL, LinIfScheduleRequestConfirmationUL, and/or LinIfGotoSleepConfirmationUL is set to CDD.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel	
BSW Parameter		**BSW Type**
LinIfChannelId		EcucIntegerParamDef
BSW Description		

This parameter holds the unique channel index value. The value shall be the same as the ComMChannelId of the ComMChannel referenced by LinIfComMNetworkHandleRef. Implementation Type: NetworkHandleType This parameter is obsolete and will be removed in future.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel	
BSW Parameter		**BSW Type**
LinIfChannelRef		EcucSymbolicNameReferenceDef
BSW Description		
Reference to the channel definition in the LIN driver.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel	
BSW Parameter		**BSW Type**
LinIfComMNetworkHandleRef		EcucSymbolicNameReferenceDef
BSW Description		
Unique handle to identify one LIN network. Reference to one of the network handles configured for the ComM.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel	
BSW Parameter		**BSW Type**
LinIfFrame		EcucParamConfContainerDef
BSW Description		
Generic container for all types of LIN frames.		
M2 Template	**M2 Description**	
TPS_SYST	One frame can be triggered 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.	

M2 Parameter	
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::LinFrameTriggering	
Mapping Rule	**Mapping Type**
Create container for each LinFrameTriggering aggregated by the PhysicalChannel representing the regarded LIN channel.	full

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame	
BSW Parameter		**BSW Type**
LinIfChecksumType		EcucEnumerationParamDef
BSW Description		
Type of checksum that the frame is using.		
M2 Template	**M2 Description**	
TPS_SYST	Type of checksum that the frame is using.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::LinFrameTriggering.linChecksum		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame	
BSW Parameter		**BSW Type**
LinIfFixedFrameSdu		EcucParamConfContainerDef
BSW Description		
In case this is a fixed frame this is the SDU (response). This container represents an eight byte array. The Byte order shall be MSB first.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfFixedFrameSdu	
BSW Parameter		**BSW Type**
LinIfFixedFrameSduByte		EcucParamConfContainerDef
BSW Description		
This container represents a byte within the 8 byte array.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfFixedFrameSdu/LinIfFixedFrameSduByte

BSW Parameter			**BSW Type**
LinIfFixedFrameSduBytePos			EcucIntegerParamDef
BSW Description			
Index of the Byte in the SDU (response) 8 byte array.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfFixedFrameSdu/LinIfFixedFrameSduByte		
BSW Parameter			**BSW Type**
LinIfFixedFrameSduByteVal			EcucIntegerParamDef
BSW Description			
Byte value in the SDU (response) 8-byte array.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame		
BSW Parameter			**BSW Type**
LinIfFrameType			EcucEnumerationParamDef
BSW Description			
Type of frame/slot. A sporadic slot may be used by a set of unconditional frames in the role of substitution frames.			
M2 Template	**M2 Description**		
TPS_SYST			
M2 Parameter			
Mapping Rule			**Mapping Type**
see details in EnumerationLiteralDef descritpions			full

BSW Module	**BSW Context**	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame	
BSW Parameter		**BSW Type**
LinIfLength		EcucIntegerParamDef
BSW Description		
Length of the LIN SDU in bytes. This parameter is obsolete and will be removed in future.		
M2 Template	**M2 Description**	
TPS_SYST	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).	

M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::Frame.frameLength	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame	
BSW Parameter		**BSW Type**
LinIfPduDirection		EcucChoiceContainerDef
BSW Description		
Direction of the frame		
M2 Template	**M2 Description**	
TPS_SYST	LIN specific attributes to the FrameTriggering	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::LinFrameTriggering		
Mapping Rule		**Mapping Type**
Create container for each existing LinFrame.		full

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfPduDirection	
BSW Parameter		**BSW Type**
LinIfInternalPdu		EcucParamConfContainerDef
BSW Description		
Represents a Diagnostic or Configuration frame : no Message ID (no PduId).		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfPduDirection	
BSW Parameter		**BSW Type**
LinIfRxPdu		EcucParamConfContainerDef
BSW Description		
represents a received PDU/frame		
M2 Template	**M2 Description**	
TPS_SYST	This relationship allows to specify explicitly which ISignals are received/sent by the connected ECU on the connected channel.	
M2 Parameter		
Fibex::Fibex4Lin::LinCommunication::LinFrameTriggering.framePort		
Mapping Rule		**Mapping Type**
Create container if the regarded LinFrameTriggering in the ECU Extract contains a reference to an "in" FramePort		full

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfPduDirection/LinIfRxPdu	
BSW Parameter		**BSW Type**

LinIfRxIndicationUL	EcucFunctionNameDef
BSW Description	
This parameter defines the name of the <User_RxIndication>. This parameter depends on the parameter LinIfUserRxIndicationUL. If LinIfUserRxIndicationUL equals PDUR, the name of the <User_RxIndication> is fixed. If LinIfUserRxIndicationUL equals CDD, the name of the <User_RxIndication> is selectable.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfPduDirection/LinIfRxPdu
BSW Parameter	**BSW Type**
LinIfRxPduRef	EcucReferenceDef
BSW Description	
Reference to the PDU that is received in this frame.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfPduDirection/LinIfRxPdu
BSW Parameter	**BSW Type**
LinIfUserRxIndicationUL	EcucEnumerationParamDef
BSW Description	
This parameter defines the upper layer (UL) module to which the indication of the successfully received LinIfRxPdu has to be routed via <User_RxIndication>.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfPduDirection
BSW Parameter	**BSW Type**
LinIfSlaveToSlavePdu	EcucParamConfContainerDef
BSW Description	
Represents a slave-to-slave PDU/frame. Master does only send the header but doesn't receive the response.	
M2 Template	**M2 Description**

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfPduDirection	
BSW Parameter		**BSW Type**
LinIfTxPdu		EcucParamConfContainerDef
BSW Description		
represents a transmitted PDU/frame		
M2 Template	**M2 Description**	
TPS_SYST	This relationship allows to specify explicitly which ISignals are received/sent by the connected ECU on the connected channel.	
M2 Parameter		
Fibex::Fibex4Lin::LinCommunication::LinFrameTriggering.framePort		
Mapping Rule		**Mapping Type**
Create container if the regarded LinFrameTriggering in the ECU Extract contains a reference to an "out" FramePort		full

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfPduDirection/LinIfTxPdu	
BSW Parameter		**BSW Type**
LinIfTxConfirmationUL		EcucFunctionNameDef
BSW Description		
This parameter defines the name of the <User_TxConfirmation>. This parameter depends on the parameter LinIfUserTxUL. If LinIfUserTxUL equals PDUR, the name of the <User_TxConfirmation> is fixed. If LinIfUserTxUL equals CDD, the name of the <User_TxConfirmation> is selectable.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfPduDirection/LinIfTxPdu	
BSW Parameter		**BSW Type**
LinIfTxPduId		EcucIntegerParamDef
BSW Description		
Identifier of the Pdu for the upper layer.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfPduDirection/LinIfTxPdu	
BSW Parameter		**BSW Type**
LinIfTxPduRef		EcucReferenceDef
BSW Description		
Reference to the PDU that is transmitted in this frame.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfPduDirection/LinIfTxPdu	
BSW Parameter		**BSW Type**
LinIfTxTriggerTransmitUL		EcucFunctionNameDef
BSW Description		
This parameter defines the name of the <User_TriggerTransmit>. This parameter depends on the parameter LinIfUserTxUL. If LinIfUserTxUL equals PDUR, the name of the <User_TriggerTransmit> is fixed. If LinIfUserTxUL equals CDD, the name of the <User_TriggerTransmit> is selectable.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfPduDirection/LinIfTxPdu	
BSW Parameter		**BSW Type**
LinIfUserTxUL		EcucEnumerationParamDef
BSW Description		
This parameter defines the upper layer (UL) module to which the trigger of the transmitted LinTxPdu (via the <User_TriggerTransmit>) or the confirmation of the successfully transmitted LinTxPdu has to be routed (via the <User_TxConfirmation>).		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame	
BSW Parameter		**BSW Type**
LinIfPid		EcucIntegerParamDef
BSW Description		
Protected ID of the LIN frame. There is no reason to calculate the Parity in run-time.		

M2 Template	M2 Description	
TPS_SYST	To describe a frames identifier on the communication system, usualy with a fixed identifierValue.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::LinFrameTriggering.identifier		
Mapping Rule		**Mapping Type**
parity needs to be calculated and added based on the identifier value specified in FrameTriggering		full

BSW Module	BSW Context		
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame		
BSW Parameter		**BSW Type**	
LinIfSubstitutionFrames		EcucParamConfContainerDef	
BSW Description			
List of sporadic frames that can be sent in a sporadic frame slot.			
M2 Template	**M2 Description**		
TPS_SYST			
M2 Parameter			
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::LinSporadicFrame.substitutedFrame,SystemTemplate::Fibex::Fibex4Lin::LinCommunication::LinEventTriggeredFrame.linUnconditional Frame			
Mapping Rule			**Mapping Type**
emulate reference from System Description			full

BSW Module	BSW Context		
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfSubstitutionFrames		
BSW Parameter		**BSW Type**	
LinIfFramePriority		EcucIntegerParamDef	
BSW Description			
Priority of sporadic frame.			
M2 Template	**M2 Description**		
TPS_SYST	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.		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::LinSporadicFrame.substitutedFrame			
Mapping Rule			**Mapping Type**
In the System Description the priority is described by the Order of the UnconditionalFrames			full

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfFrame/LinIfSubstitutionFrames	
BSW Parameter		**BSW Type**
LinIfSubstitutionFrameRef		EcucReferenceDef
BSW Description		
Reference to an unconditional Frame that is used as sporadic frame.		
M2 Template	**M2 Description**	
M2 Parameter		

Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**		
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel		
BSW Parameter		**BSW Type**	
LinIfGotoSleepConfirmationUL		EcucEnumerationParamDef	
BSW Description			
This parameter defines the upper layer (UL) module to which the confirmation of the goto-sleep command shall be sent.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	**BSW Context**		
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel		
BSW Parameter		**BSW Type**	
LinIfMaster		EcucParamConfContainerDef	
BSW Description			
Each Master can only be connected to one physical channel. This could be compared to the Node parameter in a LDF file.			
M2 Template	**M2 Description**		
TPS_SYST	Describing the properties of the refering ecu as a LIN master.		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinMaster			
Mapping Rule			**Mapping Type**
Create container if the regarded ECU contains a CommunicationController that is defined as a LinMaster. In the System Template the LinMaster is connected to the LinChannel via a CommunicationConnector.			full

BSW Module	**BSW Context**		
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfMaster		
BSW Parameter		**BSW Type**	
LinIfClusterTimeBase		EcucFloatParamDef	
BSW Description			
Defines a time-base for one LIN cluster in seconds (normally 0.002, 0.005 or 0.010s). This parameter is obsolete and will be removed in future.			
M2 Template	**M2 Description**		
TPS_SYST	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."		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinMaster.timeBase			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfMaster	
BSW Parameter		**BSW Type**
LinIfJitter		EcucFloatParamDef
BSW Description		
The jitter specifies the differences between the maximum and minimum delay from time base tick to the header sending start point in seconds.		
M2 Template	**M2 Description**	
TPS_SYST	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). Unit: seconds	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinMaster.timeBaseJitter		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel	
BSW Parameter		**BSW Type**
LinIfMaxFrameCnt		EcucIntegerParamDef
BSW Description		
Maximum number of Frames. This parameter is needed only in case of post-build loadable implementation using static memory allocation.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel	
BSW Parameter		**BSW Type**
LinIfScheduleRequestConfirmationUL		EcucEnumerationParamDef
BSW Description		
This parameter defines the upper layer (UL) module to which the confirmation of the successfully performed schedule table change shall be sent.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel	
BSW Parameter		**BSW Type**
LinIfScheduleTable		EcucParamConfContainerDef
BSW Description		

Describes a schedule table. Each LinIfChannel may have several schedule tables. Each schedule table can only be connected to one channel. The SHORT-NAME of the LinIfScheduleTable container represents the symbolic name of the schedule table.		
M2 Template	**M2 Description**	
TPS_SYST	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.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::LinScheduleTable		
Mapping Rule		**Mapping Type**
Create container for each ScheduleTable that is defined for this channel.		full

BSW Module	**BSW Context**	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfScheduleTable	
BSW Parameter		**BSW Type**
LinIfEntry		EcucParamConfContainerDef
BSW Description		
Describes an entry in the schedule table (also known as Frame Slot).		
M2 Template	**M2 Description**	
TPS_SYST	Specification of a sending behavior where the transmission order is predefined, e.g. used on LIN buses	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::ScheduleTableEntry		
Mapping Rule		**Mapping Type**
Each RelativelyScheduledTiming element in the System Description requires the creation of a LinIfEntry. RelativelyScheduledTiming.scheduleTablle decides to which schedule table the LinIfEntry belongs.		full

BSW Module	**BSW Context**	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfScheduleTable/LinIfEntry	
BSW Parameter		**BSW Type**
LinIfCollisionResolvingRef		EcucReferenceDef
BSW Description		
Reference to the schedule table, which resolves the collision. This parameter is only used if the referenced frames are event triggered frames.		
M2 Template	**M2 Description**	
TPS_SYST	Reference to the schedule table, which resolves a collision.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::LinEventTriggeredFrame.collisionResolvingSchedule		
Mapping Rule		**Mapping Type**
Emulate the reference from the System Description.		full

BSW Module	**BSW Context**	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfScheduleTable/LinIfEntry	
BSW Parameter		**BSW Type**
LinIfDelay		EcucFloatParamDef
BSW Description		
Delay to next entry in schedule table in seconds.		

M2 Template	M2 Description	
TPS_SYST	RRelative delay between this tableEntry and the start of the successor in the schedule table in seconds.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::ScheduleTableEntry.delay		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfScheduleTable/LinIfEntry	
BSW Parameter		**BSW Type**
LinIfEntryIndex		EcucIntegerParamDef
BSW Description		
Position of the Frame Entry in the Schedule Table. The first entry index in the schedule table is 0.		
M2 Template	**M2 Description**	
TPS_SYST	Relative position in the schedule table. The first entry index in the schedule table is 0.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::ScheduleTableEntry.positionInTable		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfScheduleTable/LinIfEntry	
BSW Parameter		**BSW Type**
LinIfFrameRef		EcucReferenceDef
BSW Description		
Reference to the frames that belong to this schedule table entry.		
M2 Template	**M2 Description**	
TPS_SYST	Specifies the LinFrame that will be transmitted in this frame slot.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::ApplicationEntry.frameTriggering		
Mapping Rule		**Mapping Type**
Emulate reference from the System Description		full

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfScheduleTable	
BSW Parameter		**BSW Type**
LinIfResumePosition		EcucEnumerationParamDef
BSW Description		
Defines where a schedule table shall proceed in case it has been interrupted by a RUN_ONCE table.		
M2 Template	**M2 Description**	
TPS_SYST	Defines, where a schedule table shall be proceeded in case if it has been interrupted by a run-once table or MRF/SRF.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::LinScheduleTable.resumePosition		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfScheduleTable	
BSW Parameter		**BSW Type**
LinIfRunMode		EcucEnumerationParamDef
BSW Description		
The schedule table can be executed in two different modes.		
M2 Template	**M2 Description**	
TPS_SYST	The schedule table can be executed in two different modes.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::LinScheduleTable.runMode		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfScheduleTable	
BSW Parameter		**BSW Type**
LinIfScheduleMode		EcucEnumerationParamDef
BSW Description		
The schedule table can be executed in three different modes. This parameter is obsolete and will be removed in future.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfScheduleTable	
BSW Parameter		**BSW Type**
LinIfScheduleTableIndex		EcucIntegerParamDef
BSW Description		
This is the unique index used by upper layers to identify a schedule. Note that the NULL_SCHEDULE for each channel must have index 0.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfScheduleTable	
BSW Parameter		**BSW Type**
LinIfScheduleTableName		EcucStringParamDef
BSW Description		
Optional schedule name used to cross-reference with a LDF. This parameter shall always be accompanied by LINIF_SCHEDULE_INDEX. This parameter is obsolete and will be removed in future.		

M2 Template	M2 Description	
TPS_SYST	Use longName to generate a name for the context element, which enables it to be ** .	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinCommunication::LinScheduleTable		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context		
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel		
BSW Parameter		**BSW Type**	
LinIfSlave		EcucParamConfContainerDef	
BSW Description			
The Node attributes of the Slaves are provided with these parameter. This container is obsolete and will be removed in future.			
M2 Template	**M2 Description**		
TPS_SYST	Describing the properties of the refering ecu as a LIN slave.		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinSlave			
Mapping Rule			**Mapping Type**
Create container if the regarded ECU contains a CommunicationController that is defined as a LinSlave. In the System Template the LinSlave is connected to the LinChannel via a CommunicationConnector			full

BSW Module	BSW Context		
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfSlave		
BSW Parameter		**BSW Type**	
LinIfConfiguredNad		EcucIntegerParamDef	
BSW Description			
Definition of the initial node address			
M2 Template	**M2 Description**		
TPS_SYST	To distinguish LIN slaves that are used twice or more within the same cluster.		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinSlave.configuredNad			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	BSW Context		
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfSlave		
BSW Parameter		**BSW Type**	
LinIfFunctionId		EcucIntegerParamDef	
BSW Description			
LIN function ID			
M2 Template	**M2 Description**		
TPS_SYST	LIN function ID		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinSlave.functionId			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfSlave	
BSW Parameter		**BSW Type**
LinIfProtocolVersion		EcucStringParamDef
BSW Description		
Defines the LIN Protocol version which is used by the slave.		
M2 Template	**M2 Description**	
TPS_SYST	Version specifier for a communication protocol	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinSlave		
Mapping Rule		**Mapping Type**
Use the protocolVersion attribute.		full

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfSlave	
BSW Parameter		**BSW Type**
LinIfSupplierId		EcucIntegerParamDef
BSW Description		
LIN Supplier ID		
M2 Template	**M2 Description**	
TPS_SYST	LIN Supplier ID	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinSlave.supplierId		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfSlave	
BSW Parameter		**BSW Type**
LinIfVariant		EcucIntegerParamDef
BSW Description		
Specifies the Variant ID		
M2 Template	**M2 Description**	
TPS_SYST	Specifies the Variant ID	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinSlave.variantId		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel	
BSW Parameter		**BSW Type**
LinIfStartupState		EcucEnumerationParamDef
BSW Description		
Defines the state of each LIN channel after startup		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

	local

BSW Module	BSW Context		
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel		
BSW Parameter		**BSW Type**	
LinIfTransceiverDrvConfig		EcucParamConfContainerDef	
BSW Description			
This container contains the configuration parameters of each underlying LIN Transceiver Driver.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfTransceiverDrvConfig		
BSW Parameter		**BSW Type**	
LinIfTrcvIdRef		EcucSymbolicNameReferenceDef	
BSW Description			
Logical handle of the underlying LIN transceiver to be served by the LIN Interface.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	BSW Context		
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel/LinIfTransceiverDrvConfig		
BSW Parameter		**BSW Type**	
LinIfTrcvWakeupNotification		EcucBooleanParamDef	
BSW Description			
Selects whether wakeup indication notification is supported. True: Enabled False: Disabled This parameter is obsolete and will be removed in future.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context	
LinIf	LinIf/LinIfGlobalConfig/LinIfChannel	
BSW Parameter		**BSW Type**

LinIfWakeupConfirmationUL	EcucEnumerationParamDef
BSW Description	
This parameter defines the upper layer (UL) module to which the confirmation of the wake-up shall be sent.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**
LinIf	LinIf/LinIfGlobalConfig
BSW Parameter	**BSW Type**
LinIfTimeBase	EcucFloatParamDef
BSW Description	
Defines the interval of calls to main functions in seconds.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

D.5.3 LinNm Mapping

BSW Module	**BSW Context**
LinNm	LinNm
BSW Parameter	**BSW Type**
LinNmGlobalConfig	EcucParamConfContainerDef
BSW Description	
This container contains the global configuration parameter of the LinNm.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**
LinNm	LinNm/LinNmGlobalConfig
BSW Parameter	**BSW Type**
LinNmBusSynchronizationEnabled	EcucBooleanParamDef
BSW Description	
Pre-processor switch for enabling bus synchronization support of the LinNm. This feature is required for NM Coordinator nodes only.	
M2 Template	**M2 Description**
TPS_SYST	Enables bus synchronization support.
M2 Parameter	
SystemTemplate::NetworkManagement::NmEcu.nmBusSynchronizationEnabled	

Mapping Rule	Mapping Type
1:1 mapping	full

BSW Module	BSW Context	
LinNm	LinNm/LinNmGlobalConfig	
BSW Parameter		**BSW Type**
LinNmChannelConfig		EcucParamConfContainerDef
BSW Description		
This container contains the channel specific configuration parameter of the LinNm.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
LinNm	LinNm/LinNmGlobalConfig/LinNmChannelConfig	
BSW Parameter		**BSW Type**
LinNmComMNetworkHandleRef		EcucSymbolicNameReferenceDef
BSW Description		
This reference points to the unique channel defined by the ComMChannel and provides access to the unique channel index value in ComMChannelId.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
LinNm	LinNm/LinNmGlobalConfig	
BSW Parameter		**BSW Type**
LinNmComControlEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling the Communication Control support.		
M2 Template	**M2 Description**	
TPS_SYST	Enables the Communication Control support.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmComControlEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
LinNm	LinNm/LinNmGlobalConfig	
BSW Parameter		**BSW Type**
LinNmComUserDataSupport		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling the NM COM user data support		

M2 Template	M2 Description	
M2 Parameter		
Mapping Rule		**Mapping Type**
		full

BSW Module	BSW Context	
LinNm	LinNm/LinNmGlobalConfig	
BSW Parameter		**BSW Type**
LinNmCoordinatorSyncSupport		EcucBooleanParamDef
BSW Description		
Enables/disables the coordinator synchronization support.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
LinNm	LinNm/LinNmGlobalConfig	
BSW Parameter		**BSW Type**
LinNmDevErrorDetect		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling development error detection support.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
LinNm	LinNm/LinNmGlobalConfig	
BSW Parameter		**BSW Type**
LinNmNodeDetectionEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling the Node Detection feature.		
M2 Template	**M2 Description**	
TPS_SYST	Enables the Request Repeat Message Request support. Only valid if nmNodeIdEnabled is set to true.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmNodeDetectionEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context
LinNm	LinNm/LinNmGlobalConfig

BSW Parameter		BSW Type	
LinNmNodeIdEnabled		EcucBooleanParamDef	
BSW Description			
Pre-processor switch for enabling transmission of the source node identifier in NM messages.			
M2 Template	**M2 Description**		
TPS_SYST	Enables the source node identifier.		
M2 Parameter			
SystemTemplate::NetworkManagement::NmEcu.nmNodeIdEnabled			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	BSW Context		
LinNm	LinNm/LinNmGlobalConfig		
BSW Parameter		**BSW Type**	
LinNmPassiveModeEnabled		EcucBooleanParamDef	
BSW Description			
Pre-processor switch for enabling support of the Passive Mode of the LinNm.			
M2 Template	**M2 Description**		
TPS_SYST	Enables support of the Passive Mode.		
M2 Parameter			
SystemTemplate::NetworkManagement::NmEcu.nmPassiveModeEnabled			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	BSW Context		
LinNm	LinNm/LinNmGlobalConfig		
BSW Parameter		**BSW Type**	
LinNmRemoteSleepIndicationEnabled		EcucBooleanParamDef	
BSW Description			
Pre-processor switch for enabling Remote Sleep Indication support. This feature is required for NM Coordinator nodes only.			
M2 Template	**M2 Description**		
TPS_SYST	Switch for enabling remote sleep indication support.		
M2 Parameter			
SystemTemplate::NetworkManagement::NmEcu.nmRemoteSleepIndEnabled			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	BSW Context		
LinNm	LinNm/LinNmGlobalConfig		
BSW Parameter		**BSW Type**	
LinNmStateChangeIndEnabled		EcucBooleanParamDef	
BSW Description			
Pre-processor switch for enabling the Network Management state change notification.			
M2 Template	**M2 Description**		
TPS_SYST	Enables the CAN Network Management state change notification.		
M2 Parameter			
SystemTemplate::NetworkManagement::NmEcu.nmStateChangeIndEnabled			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	BSW Context	
LinNm	LinNm/LinNmGlobalConfig	
BSW Parameter		**BSW Type**
LinNmSynchronizationPointEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling the Synchronize NM feature.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
LinNm	LinNm/LinNmGlobalConfig	
BSW Parameter		**BSW Type**
LinNmUserDataEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling User Data support.		
M2 Template	**M2 Description**	
TPS_SYST	Switch for enabling user data support.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmUserDataEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
LinNm	LinNm/LinNmGlobalConfig	
BSW Parameter		**BSW Type**
LinNmVersionInfoApi		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling version info API support.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

D.5.4 LinTp Mapping

BSW Module	BSW Context	
LinTp	LinTp	
BSW Parameter		**BSW Type**
LinTpGeneral		EcucParamConfContainerDef
BSW Description		
Container that holds all LIN transport protocol general parameters.		
M2 Template	**M2 Description**	
M2 Parameter		

Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**	
LinTp	LinTp/LinTpGeneral	
BSW Parameter		**BSW Type**
LinTpChangeParameterApi		EcucBooleanParamDef
BSW Description		
This parameter, if set to true, enables the LinTp_ChangeParameterRequest Api for this Module.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
LinTp	LinTp/LinTpGeneral	
BSW Parameter		**BSW Type**
LinTpVersionInfoApi		EcucBooleanParamDef
BSW Description		
Switches the LinTp_GetVersionInfo function ON or OFF.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
LinTp	LinTp	
BSW Parameter		**BSW Type**
LinTpGlobalConfig		EcucParamConfContainerDef
BSW Description		
This container contains the global configuration parameters of the LinTp. It is a MultipleConfigurationContainer, i.e. this container and its sub-containers exist once per configuration set.		
M2 Template	**M2 Description**	
TPS_SYST	TP Node (Sender or Receiver) provides the TP Address and the connection to the Topology description.	
M2 Parameter		
SystemTemplate::TransportProtocols::LinTpNode		
Mapping Rule		**Mapping Type**
Create container if the regarded ECU is a LinTpNode.		full

BSW Module	**BSW Context**	
LinTp	LinTp/LinTpGlobalConfig	
BSW Parameter		**BSW Type**
LinTpChannelConfig		EcucParamConfContainerDef

BSW Description		
This container contains the channel specific configuration parameters of LinTp.		
M2 Template	**M2 Description**	
TPS_SYST	Reference to the IPdu that is segmented by the Transport Protocol.	
M2 Parameter		
SystemTemplate::TransportProtocols::LinTpConnection.linTpNSdu		
Mapping Rule		**Mapping Type**
Create container for each NSdu that is received by the regarded ECU.		full

BSW Module	BSW Context		
LinTp	LinTp/LinTpGlobalConfig/LinTpChannelConfig		
BSW Parameter		**BSW Type**	
LinTpChannelRef		EcucSymbolicNameReferenceDef	
BSW Description			
Index of the channel this LinTp channel belongs to.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	BSW Context		
LinTp	LinTp/LinTpGlobalConfig/LinTpChannelConfig		
BSW Parameter		**BSW Type**	
LinTpDropNotRequestedNad		EcucBooleanParamDef	
BSW Description			
Configures if TP Frames of not requested LIN-Slaves are dropped or not. TRUE: Drop TP Frames of not requested LIN-Slaves FALSE: Keep TP Frames of not requested LIN-Slaves			
M2 Template	**M2 Description**		
TPS_SYST	Configures if TP Frames of not requested LIN-Slaves are dropped or not.		
M2 Parameter			
SystemTemplate::TransportProtocols::LinTpConnection.dropNotRequestedNad			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	BSW Context	
LinTp	LinTp/LinTpGlobalConfig/LinTpChannelConfig	
BSW Parameter		**BSW Type**
LinTpScheduleChangeDiag		EcucBooleanParamDef
BSW Description		
Enables or disables the call of BswM_LinTp_RequestMode() to diagnostic request/response schedule. false: BswM is not called true: BswM is called		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context	
LinTp	LinTp/LinTpGlobalConfig	
BSW Parameter		**BSW Type**
LinTpMaxNumberOfRespPendingFrames		EcucIntegerParamDef
BSW Description		
Configures the maximum number of allowed response pending frames.		
M2 Template	**M2 Description**	
TPS_SYST	Configures the maximum number of allowed response pending frames.	
M2 Parameter		
SystemTemplate::TransportProtocols::LinTpNode.maxNumberOfRespPendingFrames		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
LinTp	LinTp/LinTpGlobalConfig	
BSW Parameter		**BSW Type**
LinTpMaxRxNSduCnt		EcucIntegerParamDef
BSW Description		
Maximum number of NSdus. This parameter is needed only in case of post-build loadable implementation using static memory allocation.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
LinTp	LinTp/LinTpGlobalConfig	
BSW Parameter		**BSW Type**
LinTpMaxTxNSduCnt		EcucIntegerParamDef
BSW Description		
Maximum number of NSdus. This parameter is needed only in case of post-build loadable implementation using static memory allocation.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
LinTp	LinTp/LinTpGlobalConfig	
BSW Parameter		**BSW Type**

LinTpNumberOfRxNSdu		EcucIntegerParamDef
BSW Description		
Number of transport protocol messages that can be received for all channels this node is connected to. This parameter is obsolete and will be removed in future.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
LinTp	LinTp/LinTpGlobalConfig	
BSW Parameter		**BSW Type**
LinTpNumberOfTxNSdu		EcucIntegerParamDef
BSW Description		
Number of transport protocol messages that can be transmitted for all channels this node is connected to. This parameter is obsolete and will be removed in future.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
LinTp	LinTp/LinTpGlobalConfig	
BSW Parameter		**BSW Type**
LinTpP2Max		EcucFloatParamDef
BSW Description		
P2*max timeout when a response pending frame is expected in seconds. Note that the minimum value of LinTpP2Max shall be more than or equal to the value of LinTpP2Timing.		
M2 Template	**M2 Description**	
TPS_SYST	After reception of a response pending frame the P2 timeout counter is reloaded with the timeout time P2max.	
M2 Parameter		
SystemTemplate::TransportProtocols::LinTpNode.p2Max		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
LinTp	LinTp/LinTpGlobalConfig	
BSW Parameter		**BSW Type**
LinTpP2Timing		EcucFloatParamDef
BSW Description		
Definition of the P2max timeout observation parameter in seconds.		

M2 Template	M2 Description	
TPS_SYST	P2 timeout observation parameter.	
M2 Parameter		
SystemTemplate::TransportProtocols::LinTpNode.p2Timing		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context		
LinTp	LinTp/LinTpGlobalConfig		
BSW Parameter		**BSW Type**	
LinTpRxNSdu		EcucParamConfContainerDef	
BSW Description			
This container exists once for each received N-SDU on any channel the node is connected to.			
M2 Template	**M2 Description**		
TPS_SYST	Reference to the IPdu that is segmented by the Transport Protocol.		
M2 Parameter			
SystemTemplate::TransportProtocols::LinTpConnection.linTpNSdu			
Mapping Rule			**Mapping Type**
Create container for each NSdu that is received by the regarded ECU.			full

BSW Module	BSW Context		
LinTp	LinTp/LinTpGlobalConfig/LinTpRxNSdu		
BSW Parameter		**BSW Type**	
LinTpDl		EcucIntegerParamDef	
BSW Description			
Data Length Code of this RxNsdu. In case of variable length message, this value indicates the minimum data length. Range of minimum length is 1 to 4095. Note that this is not relevant for Tx. The reason for this is to have identical structures for Tx and Rx. This parameter is obsolete and will be removed in future.			
M2 Template	**M2 Description**		
TPS_SYST	The size of the IPDU in bits.		
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreCommunication::Pdu.length			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	BSW Context	
LinTp	LinTp/LinTpGlobalConfig/LinTpRxNSdu	
BSW Parameter		**BSW Type**
LinTpNcr		EcucFloatParamDef
BSW Description		
Value in seconds of the N_Cr timeout. N_Cr is the time until reception of the next Consecutive Frame N_PDU.		
M2 Template	**M2 Description**	

TPS_SYST	This attribute 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.	
M2 Parameter		
SystemTemplate::TransportProtocols::LinTpConnection.timeoutCr		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
LinTp	LinTp/LinTpGlobalConfig/LinTpRxNSdu	
BSW Parameter		**BSW Type**
LinTpRxNSduChannelRef		EcucSymbolicNameReferenceDef
BSW Description		
Index of the channel this N-SDU belongs to.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
LinTp	LinTp/LinTpGlobalConfig/LinTpRxNSdu	
BSW Parameter		**BSW Type**
LinTpRxNSduId		EcucIntegerParamDef
BSW Description		
The identifier of the Transport Protocol message. This ID will be used by upper layers to call LinTp_ChangeParameter and LinTp_CancelReceive.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
LinTp	LinTp/LinTpGlobalConfig/LinTpRxNSdu	
BSW Parameter		**BSW Type**
LinTpRxNSduNad		EcucIntegerParamDef
BSW Description		
A N-SDU transported on LIN is identified using the NAD for the specific slave.		
M2 Template	**M2 Description**	
TPS_SYST	To distinguish LIN slaves that are used twice or more within the same cluster.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinSlave.configuredNad		
Mapping Rule		**Mapping Type**
Find connection from NSdu to CommunicationController		full

BSW Module	**BSW Context**

LinTp	LinTp/LinTpGlobalConfig/LinTpRxNSdu		
BSW Parameter		**BSW Type**	
LinTpRxNSduPduRef		EcucReferenceDef	
BSW Description			
Reference to the global PDU			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
LinTp	LinTp/LinTpGlobalConfig		
BSW Parameter		**BSW Type**	
LinTpTxNSdu		EcucParamConfContainerDef	
BSW Description			
This container exists once for each transmitted N-SDU on any channel the node is connected to.			
M2 Template	**M2 Description**		
TPS_SYST	Reference to the IPdu that is segmented by the Transport Protocol.		
M2 Parameter			
SystemTemplate::TransportProtocols::LinTpConnection.linTpNSdu			
Mapping Rule			**Mapping Type**
Create container for each NSdu that is received by the regarded ECU.			full

BSW Module	**BSW Context**		
LinTp	LinTp/LinTpGlobalConfig/LinTpTxNSdu		
BSW Parameter		**BSW Type**	
LinTpMaxBufReq		EcucIntegerParamDef	
BSW Description			
This parameter defines the maximum number of times the LinTp should request upper layer for the Tx Buffer. It is also used to limit the number of retries for PduR_LinTpCopyTxData when no timer is active.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	**BSW Context**		
LinTp	LinTp/LinTpGlobalConfig/LinTpTxNSdu		
BSW Parameter		**BSW Type**	
LinTpNas		EcucFloatParamDef	
BSW Description			
Value in seconds of the N_As timeout. N_As is the time for transmission of a LIN frame (any N_PDU) on the part of the sender.			
M2 Template	**M2 Description**		
TPS_SYST	Time for transmission of the LIN frame (any N-PDU) on the sender side. Specified in seconds.		

M2 Parameter	
SystemTemplate::TransportProtocols::LinTpConnection.timeoutAs	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	BSW Context	
LinTp	LinTp/LinTpGlobalConfig/LinTpTxNSdu	
BSW Parameter		**BSW Type**
LinTpNcs		EcucFloatParamDef
BSW Description		
Value in seconds of the performance requirement of N_Cs. N_Cs is the time which elapses between the transmit request of a CF N-PDU until the transmit request of the next CF N-PDU.		
M2 Template	**M2 Description**	
TPS_SYST	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.	
M2 Parameter		
SystemTemplate::TransportProtocols::LinTpConnection.timeoutCs		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
LinTp	LinTp/LinTpGlobalConfig/LinTpTxNSdu	
BSW Parameter		**BSW Type**
LinTpTxNSduChannelRef		EcucSymbolicNameReferenceDef
BSW Description		
Index of the channel this N-SDU belongs to.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
LinTp	LinTp/LinTpGlobalConfig/LinTpTxNSdu	
BSW Parameter		**BSW Type**
LinTpTxNSduId		EcucIntegerParamDef
BSW Description		
The identifier of the Transport Protocol message. This ID will be the one that is communicated with upper layers.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context

LinTp	LinTp/LinTpGlobalConfig/LinTpTxNSdu	
BSW Parameter		**BSW Type**
LinTpTxNSduNad		EcucIntegerParamDef
BSW Description		
A N-SDU transported on LIN is identified using the NAD for the specific slave.		
M2 Template	**M2 Description**	
TPS_SYST	To distinguish LIN slaves that are used twice or more within the same cluster.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Lin::LinTopology::LinSlave.configuredNad		
Mapping Rule		**Mapping Type**
Find connection from NSdu to CommunicationController		full

BSW Module	**BSW Context**	
LinTp	LinTp/LinTpGlobalConfig/LinTpTxNSdu	
BSW Parameter		**BSW Type**
LinTpTxNSduPduRef		EcucReferenceDef
BSW Description		
Reference to the global PDU		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

D.6 Ethernet

//

D.6.1 Ethernet Driver Mapping

// //

BSW Module	**BSW Context**	
Eth	Eth	
BSW Parameter		**BSW Type**
EthConfigSet		EcucParamConfContainerDef
BSW Description		
All included containers and parameters that may be part of a multiple configuration set.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**
Eth	Eth/EthConfigSet

BSW Parameter		BSW Type	
EthCtrlConfig		EcucParamConfContainerDef	
BSW Description			
Configuration of the individual controller			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
Eth	Eth/EthConfigSet/EthCtrlConfig		
BSW Parameter		**BSW Type**	
EthCtrlEnableMii		EcucBooleanParamDef	
BSW Description			
Enables / Disables Media Independent Interface (MII) for transceiver access			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
Eth	Eth/EthConfigSet/EthCtrlConfig		
BSW Parameter		**BSW Type**	
EthCtrlEnableRxInterrupt		EcucBooleanParamDef	
BSW Description			
Enables / Disables receive interrupt			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
Eth	Eth/EthConfigSet/EthCtrlConfig		
BSW Parameter		**BSW Type**	
EthCtrlEnableTxInterrupt		EcucBooleanParamDef	
BSW Description			
Enables / Disables transmit interrupt			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context	
Eth	Eth/EthConfigSet/EthCtrlConfig	
BSW Parameter		**BSW Type**
EthCtrlIdx		EcucIntegerParamDef
BSW Description		
Specifies the instance ID of the configured controller.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Eth	Eth/EthConfigSet/EthCtrlConfig	
BSW Parameter		**BSW Type**
EthCtrlPhyAddress		EcucStringParamDef
BSW Description		
Specifies the unique 48-bit physical address (MAC address) of the controller in network byte order. Regular Expression: [0-9a-fA-F]{2}[[:-][0-9a-fA-F]{2}]{5}		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Eth	Eth/EthConfigSet/EthCtrlConfig	
BSW Parameter		**BSW Type**
EthCtrlRxBufLenByte		EcucIntegerParamDef
BSW Description		
Limits the maximum receive buffer length (frame length) in bytes.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Eth	Eth/EthConfigSet/EthCtrlConfig	
BSW Parameter		**BSW Type**
EthCtrlTxBufLenByte		EcucIntegerParamDef
BSW Description		
Limits the maximum transmit buffer length (frame length) in bytes.		
M2 Template	**M2 Description**	
M2 Parameter		

Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**		
Eth	Eth/EthConfigSet/EthCtrlConfig		
BSW Parameter		**BSW Type**	
EthDemEventParameterRefs		EcucParamConfContainerDef	
BSW Description			
Container for the references to DemEventParameter elements which shall be invoked using the API Dem_ReportErrorStatus in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId value. The standardized errors are provided in the container and can be extended by vendor specific error references.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
Eth	Eth/EthConfigSet/EthCtrlConfig/EthDemEventParameterRefs		
BSW Parameter		**BSW Type**	
ETH_E_ACCESS		EcucSymbolicNameReferenceDef	
BSW Description			
Reference to the DemEventParameter which shall be issued when the error "Controller access failed" has occured.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
Eth	Eth/EthConfigSet/EthCtrlConfig		
BSW Parameter		**BSW Type**	
EthRxBufTotal		EcucIntegerParamDef	
BSW Description			
Configures the number of receive buffers.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**
Eth	Eth/EthConfigSet/EthCtrlConfig

BSW Parameter		BSW Type
EthTxBufTotal		EcucIntegerParamDef
BSW Description		
Configures the number of transmit buffers.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Eth	Eth	
BSW Parameter		**BSW Type**
EthGeneral		EcucParamConfContainerDef
BSW Description		
General configuration of Ethernet Driver module		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Eth	Eth/EthGeneral	
BSW Parameter		**BSW Type**
EthDevErrorDetect		EcucBooleanParamDef
BSW Description		
Enables / Disables development error detection.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Eth	Eth/EthGeneral	
BSW Parameter		**BSW Type**
EthIndex		EcucIntegerParamDef
BSW Description		
Specifies the InstanceId of this module instance. If only one instance is present it shall have the Id 0.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Eth	Eth/EthGeneral	
BSW Parameter		**BSW Type**
EthMaxCtrlsSupported		EcucIntegerParamDef
BSW Description		
Limits the total number of supported controllers.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Eth	Eth/EthGeneral	
BSW Parameter		**BSW Type**
EthUpdatePhysAddrFilter		EcucBooleanParamDef
BSW Description		
Enables/Disables optional API Eth_UpdatePhysAddrFilter.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Eth	Eth/EthGeneral	
BSW Parameter		**BSW Type**
EthVersionInfoApi		EcucBooleanParamDef
BSW Description		
Enables / Disables version info API		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

D.6.2 Ethernet Interface Mapping

BSW Module	BSW Context	
EthIf	EthIf	
BSW Parameter		**BSW Type**
EthIfConfigSet		EcucParamConfContainerDef
BSW Description		
Collecting container for all parameters with post-build configuration classes.		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**	
EthIf	EthIf/EthIfConfigSet	
BSW Parameter		**BSW Type**
EthIfController		EcucParamConfContainerDef
BSW Description		
This container contains the configuration of EthIfController.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
EthIf	EthIf/EthIfConfigSet/EthIfController	
BSW Parameter		**BSW Type**
EthIfCtrlIdx		EcucIntegerParamDef
BSW Description		
This parameter provides a zero-based consecutive index of the Ethernet Communication Controllers. Upper layer BSW modules and the EthIf itself use this index to identify a Ethernet CC.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
EthIf	EthIf/EthIfConfigSet/EthIfController	
BSW Parameter		**BSW Type**
EthIfCtrlMtu		EcucIntegerParamDef
BSW Description		
Specifies the maximum transmission unit (MTU) of the EthIfCtrl in [bytes]. Note: in case a VLAN tag is used for the EthIfCtrl, the MTU is 4 bytes smaller than the maximum payload size of an Ethernet frame which can be transmitted on the network.		
M2 Template	**M2 Description**	
TPS_SYST	This attribute specifies the maximum transmission unit in bytes.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetCommunicationController.maximumTransmissionUnit		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
EthIf	EthIf/EthIfConfigSet/EthIfController	
BSW Parameter		**BSW Type**
EthIfEthCtrlRef		EcucSymbolicNameReferenceDef
BSW Description		
Reference to a Controller, which is handled by a specific Driver. This reference is unique for the ECU.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
EthIf	EthIf/EthIfConfigSet/EthIfController	
BSW Parameter		**BSW Type**
EthIfEthTrcvRef		EcucSymbolicNameReferenceDef
BSW Description		
Reference to a Ethernet Transceiver.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
EthIf	EthIf/EthIfConfigSet/EthIfController	
BSW Parameter		**BSW Type**
EthIfMaxTxBufsTotal		EcucIntegerParamDef
BSW Description		
Limits the total number of transmit buffers.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
EthIf	EthIf/EthIfConfigSet/EthIfController	
BSW Parameter		**BSW Type**
EthIfVlanId		EcucIntegerParamDef
BSW Description		
A virtual-LAN is identified by this attribute according to IEEE 802.1Q.		
M2 Template	**M2 Description**	
TPS_SYST	A virtual-LAN is identified by this attribute according to IEEE 802.1Q. The allowed values range is from 0..4095.	
M2 Parameter		

SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::VlanConfig.vlanIdentifier	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**	
EthIf	EthIf/EthIfConfigSet	
BSW Parameter		**BSW Type**
EthIfFrameOwnerConfig		EcucParamConfContainerDef
BSW Description		
Configuration of Ethernet frame owner		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
EthIf	EthIf/EthIfConfigSet/EthIfFrameOwnerConfig	
BSW Parameter		**BSW Type**
EthIfFrameType		EcucIntegerParamDef
BSW Description		
Selects the Ethernet frame type.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
EthIf	EthIf/EthIfConfigSet/EthIfFrameOwnerConfig	
BSW Parameter		**BSW Type**
EthIfOwner		EcucIntegerParamDef
BSW Description		
Selects the owner of an Ethernet frame type. The owner is a zero based index into the callback function configuration 'EthIfRxIndicationConfig'. I.e. an Ethernet frame of type IPv4 (0x800) at index 0 will call the first callback function configured in 'EthIfRxIndicationConfig'.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
EthIf	EthIf/EthIfConfigSet	
BSW Parameter		**BSW Type**
EthIfRxIndicationConfig		EcucParamConfContainerDef

BSW Description	
Configuration of receive callback functions.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context
EthIf	EthIf/EthIfConfigSet/EthIfRxIndicationConfig
BSW Parameter	**BSW Type**
EthIfRxIndicationFunction	EcucFunctionNameDef
BSW Description	
Specifies receive indication callback function.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context
EthIf	EthIf/EthIfConfigSet
BSW Parameter	**BSW Type**
EthIfTrcvLinkStateChgConfig	EcucParamConfContainerDef
BSW Description	
Specifies link state change callback function	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context
EthIf	EthIf/EthIfConfigSet/EthIfTrcvLinkStateChgConfig
BSW Parameter	**BSW Type**
EthIfTrcvLinkStateChgFunction	EcucFunctionNameDef
BSW Description	
Specifies link state change callback function	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context

EthIf	EthIf/EthIfConfigSet		
BSW Parameter		**BSW Type**	
EthIfTxConfirmationConfig		EcucParamConfContainerDef	
BSW Description			
Configuration of transmit indication callback functions.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
EthIf	EthIf/EthIfConfigSet/EthIfTxConfirmationConfig		
BSW Parameter		**BSW Type**	
EthIfTxConfirmationFunction		EcucFunctionNameDef	
BSW Description			
Specifies transmit indication callback function			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
EthIf	EthIf		
BSW Parameter		**BSW Type**	
EthIfGeneral		EcucParamConfContainerDef	
BSW Description			
This container contains the general configuration parameters of the Ethernet Interface.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
EthIf	EthIf/EthIfGeneral		
BSW Parameter		**BSW Type**	
EthIfDevErrorDetect		EcucBooleanParamDef	
BSW Description			
Enables / Disables development error detection.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context	
EthIf	EthIf/EthIfGeneral	
BSW Parameter		**BSW Type**
EthIfEnableRxInterrupt		EcucBooleanParamDef
BSW Description		
Enables / Disables receive interrupt.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
EthIf	EthIf/EthIfGeneral	
BSW Parameter		**BSW Type**
EthIfEnableTxInterrupt		EcucBooleanParamDef
BSW Description		
Enables / Disables the transmit interrupt.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
EthIf	EthIf/EthIfGeneral	
BSW Parameter		**BSW Type**
EthIfGetBaudRate		EcucBooleanParamDef
BSW Description		
Enables / Disables GetBaudRate API.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
EthIf	EthIf/EthIfGeneral	
BSW Parameter		**BSW Type**
EthIfGetCounterState		EcucBooleanParamDef
BSW Description		
Enables / Disables GetCounterState API.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

	local

BSW Module	BSW Context		
EthIf	EthIf/EthIfGeneral		
BSW Parameter		**BSW Type**	
EthIfMainFunctionPeriod		EcucFloatParamDef	
BSW Description			
Specifies the period of main function EthIf_MainFunctionRx and EthIf_MainFunctionTx in seconds. Ethernet Interface does not require this information but the BSW scheduler.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
EthIf	EthIf/EthIfGeneral		
BSW Parameter		**BSW Type**	
EthIfMainFunctionRxTimeout		EcucFloatParamDef	
BSW Description			
Timeout in seconds after which the EthIf stops to receive frames in an EthIfMainFunctionRx period.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
EthIf	EthIf/EthIfGeneral		
BSW Parameter		**BSW Type**	
EthIfMaxTrcvsTotal		EcucIntegerParamDef	
BSW Description			
Limits the total number of transceivers.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context	
EthIf	EthIf/EthIfGeneral	
BSW Parameter		**BSW Type**
EthIfPublicCddHeaderFile		EcucStringParamDef
BSW Description		
Defines header files for callback functions which shall be included in case of CDDs. Range of characters is 1.. 32.		

M2 Template	M2 Description	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context		
EthIf	EthIf/EthIfGeneral		
BSW Parameter		**BSW Type**	
EthIfRxIndicationIterations		EcucIntegerParamDef	
BSW Description			
Maximum number of Ethernet frames per Ethernet controller polled from the Ethernet driver within EthIf_MainFunctionRx.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
EthIf	EthIf/EthIfGeneral		
BSW Parameter		**BSW Type**	
EthIfStartAutoNegotiation		EcucBooleanParamDef	
BSW Description			
Enables / Disables StartAutoNegotiation API.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
EthIf	EthIf/EthIfGeneral		
BSW Parameter		**BSW Type**	
EthIfTrcvLinkStateChgMainReload		EcucIntegerParamDef	
BSW Description			
Specifies the frequency of transceiver link state change checks in each period of main function EthIf_MainFunctionTx.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context

EthIf	EthIf/EthIfGeneral		
BSW Parameter		**BSW Type**	
EthIfVersionInfoApi		EcucBooleanParamDef	
BSW Description			
Enables / Disables version info API			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
EthIf	EthIf/EthIfGeneral		
BSW Parameter		**BSW Type**	
EthIfVersionInfoApiMacro		EcucBooleanParamDef	
BSW Description			
Enables / Disables version info API macro implementation.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

D.6.3 Service Discovery

BSW Module	**BSW Context**		
Sd	Sd		
BSW Parameter		**BSW Type**	
SdConfig		EcucParamConfContainerDef	
BSW Description			
This container contains the configuration parameters and sub containers of the AUTOSAR Service Discovery module. This container is a MultipleConfigurationContainer, i.e. this container and its sub-containers exist once per configuration set.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**	
Sd	Sd/SdConfig	
BSW Parameter		**BSW Type**
SdInstance		EcucParamConfContainerDef
BSW Description		
This container represents an instance of the SD; i.e. the SD configuration for a certain link.		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance	
BSW Parameter		**BSW Type**
SdClientService		EcucParamConfContainerDef
BSW Description		
This container specifies all parameters used by Client services.		
M2 Template	**M2 Description**	
TPS_SYST	Service instances that are consumed by the ECU that is connected via the ApplicationEndpoint to a CommunicationConnector.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::ConsumedServiceInstance		
Mapping Rule		**Mapping Type**
Create container for each existing ConsumedServiceInstance that is available in the Ecu Extract.		full

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdClientService	
BSW Parameter		**BSW Type**
SdClientCapabilityRecord		EcucParamConfContainerDef
BSW Description		
Sd uses capability records to store arbitrary name/value pairs conveying additional information about the named service. The following use cases are supported: 1) Key present, with no value (e.g. "passreq" – password required for this service) 2) Key present, with empty value (e.g. "PlugIns=" server supports plugins, but none are presently installed) 3) Key present, with non-empty value (e.g. "PlugIns=JPEG,MPEG2,MPEG4")		
M2 Template	**M2 Description**	
TPS_SYST	SD uses capability records to store arbitrary name/value pairs conveying additional information about the named service.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::SdClientConfig.capabilityRecord		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdClientService/SdClientCapabilityRecord	
BSW Parameter		**BSW Type**
SdClientServiceCapabilityRecordKey		EcucStringParamDef
BSW Description		
Defines a CapabilityRecord key.		
M2 Template	**M2 Description**	
TPS_SYST	Defines a key.	

M2 Parameter	
GenericStructure::GeneralTemplateClasses::TagWithOptionalValue::TagWithOptionalValue.key	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdClientService/SdClientCapabilityRecord	
BSW Parameter		**BSW Type**
SdClientServiceCapabilityRecordValue		EcucStringParamDef
BSW Description		
Defines the corresponding CapabilityRecord value.		
M2 Template	**M2 Description**	
TPS_SYST	Defines the corresponding value.	
M2 Parameter		
GenericStructure::GeneralTemplateClasses::TagWithOptionalValue::TagWithOptionalValue.value		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdClientService	
BSW Parameter		**BSW Type**
SdClientServiceAutoRequire		EcucBooleanParamDef
BSW Description		
If existing and set to true, this Service will be set to "required" on start.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdClientService	
BSW Parameter		**BSW Type**
SdClientServiceHandleId		EcucIntegerParamDef
BSW Description		
The HandleId by which the BswM can identify this Client Service Instance.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdClientService	
BSW Parameter		**BSW Type**
SdClientServiceId		EcucIntegerParamDef
BSW Description		

Id to identify the service. This is unique for the service interface.		
M2 Template	**M2 Description**	
TPS_SYST	Reference to a providedServiceInstance to get the instanceIdentifier information from the ProvidedServiceInstance.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::ConsumedServiceInstance.providedServiceInstance		
Mapping Rule		**Mapping Type**
Shall be derived from the ConsumedServiceInstance.providedServiceInstance reference (serviceIdentifier attribute).		full

BSW Module	**BSW Context**	
Sd	Sd/SdConfig/SdInstance/SdClientService	
BSW Parameter		**BSW Type**
SdClientServiceInstanceId		EcucIntegerParamDef
BSW Description		
Configuration parameter to specify Instance Id of the service as used in SD entries.		
M2 Template	**M2 Description**	
TPS_SYST	Reference to a providedServiceInstance to get the instanceIdentifier information from the ProvidedServiceInstance.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::ConsumedServiceInstance.providedServiceInstance		
Mapping Rule		**Mapping Type**
Shall be derived from the ConsumedServiceInstance.providedServiceInstance reference (InstanceIdentifier attribute).		full

BSW Module	**BSW Context**	
Sd	Sd/SdConfig/SdInstance/SdClientService	
BSW Parameter		**BSW Type**
SdClientServiceMajorVersion		EcucIntegerParamDef
BSW Description		
Major version number of the Service as used in the SD entries.		
M2 Template	**M2 Description**	
TPS_SYST	Major version number of the Service.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::SdClientConfig.clientServiceMajorVersion		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Sd	Sd/SdConfig/SdInstance/SdClientService	
BSW Parameter		**BSW Type**
SdClientServiceMinorVersion		EcucIntegerParamDef
BSW Description		
Minor version number of the Service as used in the SD Service Entries.		
M2 Template	**M2 Description**	
TPS_SYST	Minor version number of the Service.	
M2 Parameter		

SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::SdClientConfig.clientServiceMinor Version	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**	
Sd	Sd/SdConfig/SdInstance/SdClientService	
BSW Parameter		**BSW Type**
SdClientServiceTimerRef		EcucReferenceDef
BSW Description		
The reference of the SdClientTimer container for this service.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
Sd	Sd/SdConfig/SdInstance/SdClientService	
BSW Parameter		**BSW Type**
SdConsumedEventGroup		EcucParamConfContainerDef
BSW Description		
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.		
M2 Template	**M2 Description**	
TPS_SYST	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.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::ConsumedEventGroup		
Mapping Rule		**Mapping Type**
Create container for every existing consumedEventGroup that is aggregated by the ConsumedServiceInstance		full

BSW Module	**BSW Context**	
Sd	Sd/SdConfig/SdInstance/SdClientService/SdConsumedEventGroup	
BSW Parameter		**BSW Type**
SdClientCapabilityRecord		EcucParamConfContainerDef
BSW Description		
Sd uses capability records to store arbitrary name/value pairs conveying additional information about the named service. The following use cases are supported: 1) Key present, with no value (e.g. "passreq" – password required for this service) 2) Key present, with empty value (e.g. "PlugIns=" server supports plugins, but none are presently installed) 3) Key present, with non-empty value (e.g. "PlugIns=JPEG,MPEG2,MPEG4")		

M2 Template	M2 Description	
TPS_SYST	SD uses capability records to store arbitrary name/value pairs conveying additional information about the named service.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::SdClientConfig.capabilityRecord		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context		
Sd	Sd/SdConfig/SdInstance/SdClientService/SdConsumedEventGroup/SdClientCapabilityRecord		
BSW Parameter		**BSW Type**	
SdClientServiceCapabilityRecordKey		EcucStringParamDef	
BSW Description			
Defines a CapabilityRecord key.			
M2 Template	**M2 Description**		
TPS_SYST	Defines a key.		
M2 Parameter			
GenericStructure::GeneralTemplateClasses::TagWithOptionalValue::TagWithOptionalValue.key			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	BSW Context		
Sd	Sd/SdConfig/SdInstance/SdClientService/SdConsumedEventGroup/SdClientCapabilityRecord		
BSW Parameter		**BSW Type**	
SdClientServiceCapabilityRecordValue		EcucStringParamDef	
BSW Description			
Defines the corresponding CapabilityRecord value.			
M2 Template	**M2 Description**		
TPS_SYST	Defines the corresponding value.		
M2 Parameter			
GenericStructure::GeneralTemplateClasses::TagWithOptionalValue::TagWithOptionalValue.value			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	BSW Context		
Sd	Sd/SdConfig/SdInstance/SdClientService/SdConsumedEventGroup		
BSW Parameter		**BSW Type**	
SdConsumedEventGroupAutoRequire		EcucBooleanParamDef	
BSW Description			
If existing and set to true, this EventGroup will be set to "required" on start.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdClientService/SdConsumedEventGroup	
BSW Parameter		**BSW Type**
SdConsumedEventGroupHandleId		EcucIntegerParamDef
BSW Description		
The HandleId by which the BswM can identify this EventGroup.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdClientService/SdConsumedEventGroup	
BSW Parameter		**BSW Type**
SdConsumedEventGroupId		EcucIntegerParamDef
BSW Description		
The Eventgroup Id of this eventGroup as a unique identifier of the eventgroup in this service. This identifier is used for EventGroup entries as well.		
M2 Template	**M2 Description**	
TPS_SYST	EventGroup ID. Shall be unique within one system to allow service discovery.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::ConsumedEventGroup.eventGroup Identifier		
Mapping Rule		**Mapping Type**
1:1 mapping		local

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdClientService/SdConsumedEventGroup	
BSW Parameter		**BSW Type**
SdConsumedEventGroupMulticastActivationRef		EcucSymbolicNameReferenceDef
BSW Description		
The reference of a Routing Group in order to activate and setup the Socket Connection for Multicast Events of this EventGroup. The multicast address from the received Multicast option is setup by SoAd_RequestIpAddrAssignment. The local address is the same as for the unicast events; thus, it was sent in the UDP Endpoint option of the Subscribe EventGroup entry.		
M2 Template	**M2 Description**	
TPS_SYST	The ServiceDiscovery module is able to activate and deactivate the PDU routing for events.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::ConsumedEventGroup.routingGroup		
Mapping Rule		**Mapping Type**
This container shall be created if the CEG that is aggregated by an Application Endpoint with a multicast configuration contains a reference to the SoAdRoutingGroup and the eventGroupControlType of the SoAdRoutingGroup is set to activationMulticast.ulticast".		full

BSW Module	BSW Context

Sd	Sd/SdConfig/SdInstance/SdClientService/SdConsumedEventGroup	
BSW Parameter		**BSW Type**
SdConsumedEventGroupTcpActivationRef		EcucSymbolicNameReferenceDef
BSW Description		
The reference of the Routing Group for activation of the data path for receiving TCP events. This element is also being used for getting the IP address and port number for building the TCP endpoint option for the Subscribe EventGroup entry. If no TCP methods are used in the service, this element is also being used for setting the remote address (TCP Endpoint option referenced by the Offer Service entry) and opening the TCP connection to the server before sending the Subscribe EventGroup entry. If multiple EventGroups of the same Service Instance are subscribed the TCP connection will be shared and must be opened only once.		
M2 Template	**M2 Description**	
TPS_SYST	The ServiceDiscovery module is able to activate and deactivate the PDU routing for events.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::ConsumedEventGroup.routingGroup		
Mapping Rule		**Mapping Type**
This container shall be created if the CEG that is aggregated by an Application Endpoint with a TcpTp configuration contains a reference to the SoAdRouting Group and the eventGroupControlType of the SoAdRoutingGroup is set to activationUnicast.st".		full

BSW Module	**BSW Context**	
Sd	Sd/SdConfig/SdInstance/SdClientService/SdConsumedEventGroup	
BSW Parameter		**BSW Type**
SdConsumedEventGroupTimerRef		EcucReferenceDef
BSW Description		
The reference of the SdClientTimer container for this eventGroup.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
Sd	Sd/SdConfig/SdInstance/SdClientService/SdConsumedEventGroup	
BSW Parameter		**BSW Type**
SdConsumedEventGroupUdpActivationRef		EcucSymbolicNameReferenceDef
BSW Description		

The reference of the Routing Group for activation of the data path for receiving UDP events. This element is also being used for getting the IP address and port number for building the UDP endpoint option for the Subscribe EventGroup entry. If no UDP methods are used in the service, this element is also being used for setting the remote address (UDP Endpoint option referenced by the Offer Service entry). If multiple EventGroups of the same Service Instance are subscribed the UDP Socket Connection will be shared and must be set only once.		
M2 Template	**M2 Description**	
TPS_SYST	The ServiceDiscovery module is able to activate and deactivate the PDU routing for events.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::ConsumedEventGroup.routingGroup		
Mapping Rule		**Mapping Type**
This container shall be created if the CEG that is aggregated by an ApplicationEndpoint with UdpTp configuration contains a reference to the SoAdRouting Group and the eventGroupControlType of the SoAdRoutingGroup is set to activationUnicast.st".		full

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdClientService	
BSW Parameter		**BSW Type**
SdConsumedMethods		EcucParamConfContainerDef
BSW Description		
Container element for representing the data path for accessing the server methods.		
M2 Template	**M2 Description**	
TPS_SYST	Service instances that are provided by the ECU that is connected via the ApplicationEndpoint to a CommunicationConnector.	
M2 Parameter		
SystemTemplate::DataMapping::DataMapping.serviceInstance		
Mapping Rule		**Mapping Type**
A method is described as a ClientServerInterface of a Software Component. If DataMappings exist that map operations of a ClientServerInterface to the regarded ConsumedServiceInstance then this container needs to be created.		full

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdClientService/SdConsumedMethods	
BSW Parameter		**BSW Type**
SdClientServiceActivationRef		EcucSymbolicNameReferenceDef
BSW Description		
Reference to a SoAdRoutingGroupRef to activate/deactivate the data path for the methods.		
M2 Template	**M2 Description**	
	The ServiceDiscovery module is able to activate and deactivate the PDU routing from and to TCP/IP-sockets.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::AbstractServiceInstance.routingGroup		
Mapping Rule		**Mapping Type**
This reference shall be created if the ConsumedServiceInstance contains a reference to the SoAdRoutingGroup.		local

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdClientService/SdConsumedMethods	
BSW Parameter		**BSW Type**
SdClientServiceTcpRef		EcucSymbolicNameReferenceDef
BSW Description		
Reference to the SoAdSocketConnection representing the data path (TCP) for communication with methods. This element is also used to set the remote address of the server and to open the TCP connection.		
M2 Template	**M2 Description**	
TPS_SYST	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.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::ApplicationEndpoint		
Mapping Rule		**Mapping Type**
Shall be derived from the ApplicationEndpoint to which the ProvidedServiceInstance is aggregated.		full

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdClientService/SdConsumedMethods	
BSW Parameter		**BSW Type**
SdClientServiceUdpRef		EcucSymbolicNameReferenceDef
BSW Description		
Reference to the SoAdSocketConnection representing the data path (UDP) for communication with methods. This element is also used to set the remote address of the server.		
M2 Template	**M2 Description**	
TPS_SYST	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.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::ApplicationEndpoint		
Mapping Rule		**Mapping Type**
Shall be derived from the ApplicationEndpoint to which the ProvidedServiceInstance is aggregated.		full

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance	
BSW Parameter		**BSW Type**
SdClientTimer		EcucParamConfContainerDef
BSW Description		
This container specifies all timers used by the Service Discovery module for Client Services.		
M2 Template	**M2 Description**	
TPS_SYST	Service Discovery Client configuration.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::SdClientConfig		
Mapping Rule		**Mapping Type**
The Timing parameters can be derived from the SdClientConfig attributes and the aggregated elements RequestResponseDelay and InitialSdDelayConfig.		full

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdClientTimer	
BSW Parameter		**BSW Type**
SdClientTimerInitialFindDelayMax		EcucFloatParamDef
BSW Description		
Max value in [s] to delay randomly the transmission of a find message. This parameter is mandatory for ClientService.		
M2 Template	**M2 Description**	
TPS_SYST	Max Value in seconds to delay radomly the first offer (if aggregated by SdServerConfig) or the transmission of a find message (if aggregated by SdClientConfig).	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::InitialSdDelayConfig.initialDelayMaxValue		
Mapping Rule		**Mapping Type**
Take information from SdClientConfig.initialFindBehavior		full

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdClientTimer	
BSW Parameter		**BSW Type**
SdClientTimerInitialFindDelayMin		EcucFloatParamDef
BSW Description		
Min value in [s] to delay randomly the transmission of a find message. This parameter is mandatory for ClientService.		
M2 Template	**M2 Description**	
TPS_SYST	Min Value in seconds to delay radomly the first offer (if aggregated by SdServerConfig) or the transmission of a find message (if aggregated by SdClientConfig).	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::InitialSdDelayConfig.initialDelayMinValue		
Mapping Rule		**Mapping Type**
Take information from SdClientConfig.initialFindBehavior		full

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdClientTimer	
BSW Parameter		**BSW Type**
SdClientTimerInitialFindRepetitionsBaseDelay		EcucFloatParamDef
BSW Description		
The base delay in [s] for find repetitions. Successive finds have an exponential back off delay (1x base delay, 2x base delay, 4x base delay, ...). This parameter is mandatory for ClientService.		
M2 Template	**M2 Description**	
TPS_SYST	The base delay for offer repetitions (if aggregated by SdServerConfig) or find repetitions (if aggregated by SdClientConfig. Successive find messages have an exponential back off delay.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::InitialSdDelayConfig.initialRepetitionBaseDelay		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientTimer

BSW Parameter	BSW Type
SdClientTimerInitialFindRepetitionsMax	EcucIntegerParamDef
BSW Description	
Configuration for the maximum number of find repetitions. This parameter is mandatory for ClientService.	
M2 Template	**M2 Description**
TPS_SYST	Describes the maximum amount of offer repetitions (if aggregated by SdServerConfig) or the maximum amount of find repetitions (if aggregated by SdClientConfig).
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::InitialSdDelayConfig.initialRepetitionsMax	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientTimer
BSW Parameter	**BSW Type**
SdClientTimerRequestResponseMaxDelay	EcucFloatParamDef
BSW Description	
Maximum allowable response delay to entries received by multicast in seconds. This parameter is mandatory for ConsumedEventGroups.	
M2 Template	**M2 Description**
TPS_SYST	Maximum allowable response delay to the find message in seconds.
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::RequestResponseDelay.maxValue	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientTimer
BSW Parameter	**BSW Type**
SdClientTimerRequestResponseMinDelay	EcucFloatParamDef
BSW Description	
Minimum allowable response delay to the find message in seconds. This parameter is mandatory for ConsumedEventGroups.	
M2 Template	**M2 Description**
TPS_SYST	Minimum allowable response delay to the find message in seconds.
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::RequestResponseDelay.minValue	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdClientTimer
BSW Parameter	**BSW Type**
SdClientTimerTTL	EcucIntegerParamDef
BSW Description	
Time to live for find and subscribe messages.	
M2 Template	**M2 Description**

TPS_SYST	TTL for Request and Subscribe messages.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::SdClientConfig.ttl		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**		
Sd	Sd/SdConfig/SdInstance		
BSW Parameter		**BSW Type**	
SdInstanceHostname		EcucStringParamDef	
BSW Description			
Configuration parameter to specify the Hostname.			
M2 Template	**M2 Description**		
TPS_SYST	Defines the fully qualified domain name (FQDN) e.g. some.example.host.		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::NetworkEndpoint.fullyQualifiedDomainName			
Mapping Rule			**Mapping Type**
Shall be derived from the NetworkEndpoint.			full

BSW Module	**BSW Context**		
Sd	Sd/SdConfig/SdInstance		
BSW Parameter		**BSW Type**	
SdInstanceMulticastRxPdu		EcucParamConfContainerDef	
BSW Description			
This container specifies the received PDU.			
M2 Template	**M2 Description**		
TPS_SYST	References to the IPduPort on every ECU of the system which sends and/or receives the I-PDU.		
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreCommunication::PduTriggering.iPduPort			
Mapping Rule			**Mapping Type**
The direction of the Pdu can be derived from the PduTriggering that refers to the GeneralPurposePdu that represents the SdPdu.			full

BSW Module	**BSW Context**		
Sd	Sd/SdConfig/SdInstance/SdInstanceMulticastRxPdu		
BSW Parameter		**BSW Type**	
SdRxPduId		EcucIntegerParamDef	
BSW Description			
ID of the PDU that will be received via the API Sd_SoAdIfRxIndication().			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**

Sd	Sd/SdConfig/SdInstance/SdInstanceMulticastRxPdu		
BSW Parameter		**BSW Type**	
SdRxPduRef		EcucReferenceDef	
BSW Description			
Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
Sd	Sd/SdConfig/SdInstance		
BSW Parameter		**BSW Type**	
SdInstanceTxPdu		EcucParamConfContainerDef	
BSW Description			
This container specifies the transmitted PDU.			
M2 Template	**M2 Description**		
TPS_SYST	References to the IPduPort on every ECU of the system which sends and/or receives the I-PDU.		
M2 Parameter			
SystemTemplate::Fibex::FibexCore::CoreCommunication::PduTriggering.iPduPort			
Mapping Rule			**Mapping Type**
The direction of the Pdu can be derived from the PduTriggering that refers to the GeneralPurposePdu that represents the SdPdu.			full

BSW Module	**BSW Context**		
Sd	Sd/SdConfig/SdInstance/SdInstanceTxPdu		
BSW Parameter		**BSW Type**	
SdTxPduRef		EcucReferenceDef	
BSW Description			
Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**	
Sd	Sd/SdConfig/SdInstance	
BSW Parameter		**BSW Type**
SdInstanceUnicastRxPdu		EcucParamConfContainerDef
BSW Description		
This container specifies the received PDU.		
M2 Template	**M2 Description**	
TPS_SYST	References to the IPduPort on every ECU of the system which sends and/or receives the I-PDU.	
M2 Parameter		

SystemTemplate::Fibex::FibexCore::CoreCommunication::PduTriggering.iPduPort	
Mapping Rule	**Mapping Type**
The direction of the Pdu can be derived from the PduTriggering that refers to the GeneralPurposePdu that represents the SdPdu.	full

BSW Module	**BSW Context**	
Sd	Sd/SdConfig/SdInstance/SdInstanceUnicastRxPdu	
BSW Parameter		**BSW Type**
SdRxPduId		EcucIntegerParamDef
BSW Description		
ID of the PDU that will be received via the API Sd_SoAdIfRxIndication().		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
Sd	Sd/SdConfig/SdInstance/SdInstanceUnicastRxPdu	
BSW Parameter		**BSW Type**
SdRxPduRef		EcucReferenceDef
BSW Description		
Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
Sd	Sd/SdConfig/SdInstance	
BSW Parameter		**BSW Type**
SdServerService		EcucParamConfContainerDef
BSW Description		
This container specifies all parameters used by Server services.		
M2 Template	**M2 Description**	
TPS_SYST	Service instances that are provided by the ECU that is connected via the ApplicationEndpoint to a CommunicationConnector.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::ProvidedServiceInstance		
Mapping Rule		**Mapping Type**
Create container for each existing ProvidedServiceInstance that is available in the Ecu Extract.		full

BSW Module	**BSW Context**	
Sd	Sd/SdConfig/SdInstance/SdServerService	
BSW Parameter		**BSW Type**

SdEventHandler		EcucParamConfContainerDef	
BSW Description			
Container Element for representing an EventGroup as part of the Service Instance.			
M2 Template	**M2 Description**		
TPS_SYST	Configures the outbound application endpoint a server uses to call a clients callback. Only required if the source TpPort is not dynamically assigned.		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EventHandler			
Mapping Rule			**Mapping Type**
Create container for every existing EventHandler that is aggregated by the ProvidedServiceInstance			full

BSW Module	**BSW Context**		
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler		
BSW Parameter		**BSW Type**	
SdEventHandlerAutoAvailable		EcucBooleanParamDef	
BSW Description			
If existing and set to true, this EventGroup will be set to "available" on start.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler		
BSW Parameter		**BSW Type**	
SdEventHandlerEventGroupId		EcucIntegerParamDef	
BSW Description			
The EventGroup Id of this EventGroup as a unique identifier of the EventGroup in this service. This identifier is used for EventGroup entries as well.			
M2 Template	**M2 Description**		
TPS_SYST	Callback Service Identifier.		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::ConsumedEventGroup.eventGroupIdentifier			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	**BSW Context**		
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler		
BSW Parameter		**BSW Type**	
SdEventHandlerHandleId		EcucIntegerParamDef	
BSW Description			
The HandleId by which the BswM can identify this EventGroup.			
M2 Template	**M2 Description**		
M2 Parameter			

Mapping Rule	Mapping Type
	local

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler	
BSW Parameter		**BSW Type**
SdEventHandlerMulticast		EcucParamConfContainerDef
BSW Description		
The subcontainer including the Routing Group for Activation of Events sent over Multicast. The activation ref is also being used for identification of the related Socket Connection in order to find the Multicast Address used in the Multicast Option referenced by the Subscribe EventGroup Ack entry.		
M2 Template	**M2 Description**	
TPS_SYST	The ServiceDiscovery module is able to activate and deactivate the PDU routing for events.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EventHandler.routingGroup		
Mapping Rule		**Mapping Type**
This container shall be created if the EventHandler that is aggregated by an ApplicationEndpoint with a multicast configuration contains a reference to the SoAdRoutingGroup.		full

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler/SdEventHandler Multicast	
BSW Parameter		**BSW Type**
SdEventActivationRef		EcucSymbolicNameReferenceDef
BSW Description		
Reference to a SoAdRoutingGroup for activation of the data path for a subscribed client (start sending events after subscribe).		
M2 Template	**M2 Description**	
TPS_SYST	This attribute defines the type of a RoutingGroup. There are RoutingGroups that activate the data path for unicast or multicast events of an event group. And there are RoutingGroups that activate the data path for initial events that are triggered events that are sent out on the server side after a client got subscribed.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication::SoAdRoutingGroup.eventGroup ControlType		
Mapping Rule		**Mapping Type**
Use this reference if eventGroupControlType is set to activationUnicast, activationMulticast or activationAndTriggerUnicast.		full

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler	
BSW Parameter		**BSW Type**
SdEventHandlerMulticastThreshold		EcucIntegerParamDef
BSW Description		

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 served with unicast and as soon as the second client arrives both will be served by multicast. This does not influence the handling of initial events, which are served using unicast only.	
M2 Template	**M2 Description**
TPS_SYST	Specifies the number of subscribed clients that trigger the server to change the transmission of events to multicast.
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EventHandler.multicastThreshold	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler
BSW Parameter	**BSW Type**
SdEventHandlerTcp	EcucParamConfContainerDef
BSW Description	
The subcontainer including the Routing Groups for Activation and Trigger Transmit for Events sent over TCP. The activation ref (or triggering ref if no activation ref exists) is also being used for identification of the related socket connections in order to find the related client by iterating the SdEventHandlerTcp elements (remote address statically configured or automatically set by opening TCP connection before subscription).	
M2 Template	**M2 Description**
TPS_SYST	The ServiceDiscovery module is able to activate and deactivate the PDU routing for events.
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EventHandler.routingGroup	
Mapping Rule	**Mapping Type**
This container shall be created if the EventHandler that is aggregated by an ApplicationEndpoint with a TcpTp configuration contains a reference to the SoAdRoutingGroup.	full

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler/SdEventHandlerTcp
BSW Parameter	**BSW Type**
SdEventActivationRef	EcucSymbolicNameReferenceDef
BSW Description	
Reference to a SoAdRoutingGroup for activation of the data path for a subscribed client (start sending events after subscribe).	
M2 Template	**M2 Description**
TPS_SYST	This attribute defines the type of a RoutingGroup. There are RoutingGroups that activate the data path for unicast or multicast events of an event group. And there are RoutingGroups that activate the data path for initial events that are triggered events that are sent out on the server side after a client got subscribed.

M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication::SoAdRoutingGroup.eventGroupControlType	
Mapping Rule	**Mapping Type**
Use this reference if eventGroupControlType is set to activationUnicast, activationMulticast or activationAndTriggerUnicast.	full

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler/SdEventHandlerTcp	
BSW Parameter		**BSW Type**
SdEventTriggeringRef		EcucSymbolicNameReferenceDef
BSW Description		
Reference to a SoAdRoutingGroup that is used for triggered transmit. Triggering is needed to sent out initial events on the server side after a client got subscribed.		
M2 Template	**M2 Description**	
TPS_SYST	This attribute defines the type of a RoutingGroup. There are RoutingGroups that activate the data path for unicast or multicast events of an event group. And there are RoutingGroups that activate the data path for initial events that are triggered events that are sent out on the server side after a client got subscribed.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication::SoAdRoutingGroup.eventGroupControlType		
Mapping Rule		**Mapping Type**
Use this reference if eventGroupControlType is set to triggerUnicast or activationAndTriggerUnicast.		full

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler	
BSW Parameter		**BSW Type**
SdEventHandlerTimerRef		EcucReferenceDef
BSW Description		
The reference of the SdServerTimer container for this EventGroup.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler	
BSW Parameter		**BSW Type**
SdEventHandlerUdp		EcucParamConfContainerDef
BSW Description		

The subcontainer including the Routing Groups for Activation and Trigger Transmit for Events sent over UDP. The activation ref (or triggering ref if no activation ref exists) is also being used for identification of the related socket connections in order to set the remote address of the client or find the related client by iterating the SdEventHandlerUdp elements (remote address statically configured or automatically set by method call before subscription).		
M2 Template	**M2 Description**	
TPS_SYST	The ServiceDiscovery module is able to activate and deactivate the PDU routing for events.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EventHandler.routingGroup		
Mapping Rule		**Mapping Type**
This container shall be created if the EventHandler that is aggregated by an ApplicationEndpoint with a UdpTp configuration contains a reference to the SoAdRoutingGroup.		full

BSW Module	**BSW Context**		
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler/SdEventHandlerUdp		
BSW Parameter		**BSW Type**	
SdEventActivationRef		EcucSymbolicNameReferenceDef	
BSW Description			
Reference to a SoAdRoutingGroup for activation of the data path for a subscribed client (start sending events after subscribe).			
M2 Template	**M2 Description**		
TPS_SYST	This attribute defines the type of a RoutingGroup. There are RoutingGroups that activate the data path for unicast or multicast events of an event group. And there are RoutingGroups that activate the data path for initial events that are triggered events that are sent out on the server side after a client got subscribed.		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication::SoAdRoutingGroup.eventGroupControlType			
Mapping Rule			**Mapping Type**
Use this reference if eventGroupControlType is set to activationUnicast, activationMulticast or activationAndTriggerUnicast.			full

BSW Module	**BSW Context**	
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler/SdEventHandlerUdp	
BSW Parameter		**BSW Type**
SdEventTriggeringRef		EcucSymbolicNameReferenceDef
BSW Description		
Reference to a SoAdRoutingGroup that is used for triggered transmit. Triggering is needed to sent out initial events on the server side after a client got subscribed.		
M2 Template	**M2 Description**	
TPS_SYST	This attribute defines the type of a RoutingGroup. There are RoutingGroups that activate the data path for unicast or multicast events of an event group. And there are RoutingGroups that activate the data path for initial events that are triggered events that are sent out on the server side after a client got subscribed.	
M2 Parameter		

SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication::SoAdRoutingGroup.eventGroupControlType	
Mapping Rule	**Mapping Type**
Use this reference if eventGroupControlType is set to triggerUnicast or activationAndTriggerUnicast.	full

BSW Module	**BSW Context**	
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler	
BSW Parameter		**BSW Type**
SdServerCapabilityRecord		EcucParamConfContainerDef
BSW Description		
Sd uses capability records to store arbitrary name/value pairs conveying additional information about the named service. The following use cases are supported: 1) Key present, with no value (e.g. "passreq" – password required for this service) 2) Key present, with empty value (e.g. "PlugIns=" server supports plugins, but none are presently installed) 3) Key present, with non-empty value (e.g. "PlugIns=JPEG,MPEG2,MPEG4")		
M2 Template	**M2 Description**	
TPS_SYST	A sequence of records to store arbitrary name/value pairs conveying additional information about the named service.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::SdServerConfig.capabilityRecord		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler/SdServerCapabilityRecord	
BSW Parameter		**BSW Type**
SdServerCapabilityRecordKey		EcucStringParamDef
BSW Description		
Defines a CapabilityRecord key.		
M2 Template	**M2 Description**	
TPS_SYST	Defines a key.	
M2 Parameter		
GenericStructure::GeneralTemplateClasses::TagWithOptionalValue::TagWithOptionalValue.key		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Sd	Sd/SdConfig/SdInstance/SdServerService/SdEventHandler/SdServerCapabilityRecord	
BSW Parameter		**BSW Type**
SdServerCapabilityRecordValue		EcucStringParamDef
BSW Description		
Defines the corresponding CapabilityRecord value.		
M2 Template	**M2 Description**	

TPS_SYST	Defines the corresponding value.	
M2 Parameter		
GenericStructure::GeneralTemplateClasses::TagWithOptionalValue::TagWithOptionalValue.value		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
Sd	Sd/SdConfig/SdInstance/SdServerService	
BSW Parameter		**BSW Type**
SdProvidedMethods		EcucParamConfContainerDef
BSW Description		
Container element for representing the needed elements of the data path for the methods provided by the service.		
M2 Template	**M2 Description**	
TPS_SYST	Service instances that are provided by the ECU that is connected via the ApplicationEndpoint to a CommunicationConnector.	
M2 Parameter		
SystemTemplate::DataMapping::DataMapping.serviceInstance		
Mapping Rule		**Mapping Type**
A method is described as a ClientServerInterface of a Software Component. If DataMappings exist that map operations of a ClientServerInterface to the regarded ProvidedServiceInstance then this container needs to be created.		full

BSW Module	**BSW Context**	
Sd	Sd/SdConfig/SdInstance/SdServerService/SdProvidedMethods	
BSW Parameter		**BSW Type**
SdServerServiceActivationRef		EcucSymbolicNameReferenceDef
BSW Description		
Reference to a SoAdRoutingGroup to activated and deactivate the data path for methods of the service.		
M2 Template	**M2 Description**	
TPS_SYST	The ServiceDiscovery module is able to activate and deactivate the PDU routing from and to TCP/IP-sockets.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::AbstractServiceInstance.routingGroup		
Mapping Rule		**Mapping Type**
This reference shall be created if the ProvidedServiceInstance contains a reference to the SoAdRoutingGroup.		full

BSW Module	**BSW Context**	
Sd	Sd/SdConfig/SdInstance/SdServerService/SdProvidedMethods	
BSW Parameter		**BSW Type**
SdServerServiceTcpRef		EcucReferenceDef
BSW Description		
Reference to SoAdSocketConnectionGroup used for methods. This is used to access the local IP address and port for building the endpoint option for offers of this service.		
M2 Template	**M2 Description**	

TPS_SYST	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.
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::ApplicationEndpoint	
Mapping Rule	**Mapping Type**
Shall be derived from the ApplicationEndpoint to which the ProvidedServiceInstance is aggregated.	local

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance/SdServerService/SdProvidedMethods
BSW Parameter	**BSW Type**
SdServerServiceUdpRef	EcucReferenceDef
BSW Description	
Reference to SoAdSocketConnectionGroup used for methods. This is used to access the local IP address and port for building the endpoint option for offers of this service.	
M2 Template	**M2 Description**
TPS_SYST	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.
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::ApplicationEndpoint	
Mapping Rule	**Mapping Type**
Shall be derived from the ApplicationEndpoint to which the ProvidedServiceInstance is aggregated.	full

BSW Module	**BSW Context**
Sd	Sd/SdConfig/SdInstance/SdServerService
BSW Parameter	**BSW Type**
SdServerCapabilityRecord	EcucParamConfContainerDef
BSW Description	
Sd uses capability records to store arbitrary name/value pairs conveying additional information about the named service. The following use cases are supported: 1) Key present, with no value (e.g. "passreq" – password required for this service) 2) Key present, with empty value (e.g. "PlugIns=" server supports plugins, but none are presently installed) 3) Key present, with non-empty value (e.g. "PlugIns=JPEG,MPEG2,MPEG4")	
M2 Template	**M2 Description**
TPS_SYST	A sequence of records to store arbitrary name/value pairs conveying additional information about the named service.
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::SdServerConfig.capabilityRecord	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdServerService/SdServerCapabilityRecord	
BSW Parameter		**BSW Type**
SdServerCapabilityRecordKey		EcucStringParamDef
BSW Description		
Defines a CapabilityRecord key.		
M2 Template	**M2 Description**	
TPS_SYST	Defines a key.	
M2 Parameter		
GenericStructure::GeneralTemplateClasses::TagWithOptionalValue::TagWithOptionalValue.key		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdServerService/SdServerCapabilityRecord	
BSW Parameter		**BSW Type**
SdServerCapabilityRecordValue		EcucStringParamDef
BSW Description		
Defines the corresponding CapabilityRecord value.		
M2 Template	**M2 Description**	
TPS_SYST	Defines the corresponding value.	
M2 Parameter		
GenericStructure::GeneralTemplateClasses::TagWithOptionalValue::TagWithOptionalValue.value		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdServerService	
BSW Parameter		**BSW Type**
SdServerServiceAutoAvailable		EcucBooleanParamDef
BSW Description		
If existing and set to true, this Service will be set to "Available" on start.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdServerService	
BSW Parameter		**BSW Type**
SdServerServiceHandleId		EcucIntegerParamDef
BSW Description		
The HandleId by which the BswM can identify this Server Service Instance.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

	local

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdServerService	
BSW Parameter		**BSW Type**
SdServerServiceId		EcucIntegerParamDef
BSW Description		
Id to identify the service. This is unique for the service interface.		
M2 Template	**M2 Description**	
TPS_SYST	Service ID. Shall be unique within one system to allow service discovery.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::ProvidedServiceInstance.serviceIdentifier		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdServerService	
BSW Parameter		**BSW Type**
SdServerServiceInstanceId		EcucIntegerParamDef
BSW Description		
Configuration parameter to specify Instance Id of the Service implemented by the Server Service.		
M2 Template	**M2 Description**	
TPS_SYST	Instance identifier. Can be used for e.g. service discovery to identify the instance of the service.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::ProvidedServiceInstance.instance Identifier		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdServerService	
BSW Parameter		**BSW Type**
SdServerServiceMajorVersion		EcucIntegerParamDef
BSW Description		
Major version number of the Service as used in SD Entries.		
M2 Template	**M2 Description**	
TPS_SYST	Major version number of the Service.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::SdServerConfig.serverServiceMajor Version		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
Sd	Sd/SdConfig/SdInstance/SdServerService	
BSW Parameter		**BSW Type**
SdServerServiceMinorVersion		EcucIntegerParamDef

BSW Description	
Minor version number of the Service as used e.g. in Offer Service entries.	
M2 Template	**M2 Description**
TPS_SYST	Minor version number of the Service.
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::SdServerConfig.serverServiceMinorVersion	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerService
BSW Parameter	**BSW Type**
SdServerServiceTimerRef	EcucReferenceDef
BSW Description	
The reference of the SdServerTimer container for this service.	
M2 Template	**M2 Description**
TPS_SYST	Server configuration parameter for Service-Discovery.
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::SdServerConfig	
Mapping Rule	**Mapping Type**
The timing parameters can be derived from the SdServerConfig attributes and the aggregated elements RequestResponseDelay and InitialSdDelayConfig.	full

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance
BSW Parameter	**BSW Type**
SdServerTimer	EcucParamConfContainerDef
BSW Description	
This container specifies all timers used by the Service Discovery module for Server Services.	
M2 Template	**M2 Description**
TPS_SYST	Server configuration parameter for Service-Discovery.
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::SdServerConfig	
Mapping Rule	**Mapping Type**
The Timing parameters can be derived from the SdServerConfig attributes and the aggregated elements RequestResponseDelay and InitialSdDelayConfig.	full

BSW Module	BSW Context
Sd	Sd/SdConfig/SdInstance/SdServerTimer
BSW Parameter	**BSW Type**
SdServerTimerInitialOfferDelayMax	EcucFloatParamDef
BSW Description	
Max value in [s] to delay randomly the first offer. This parameter is mandatory for ServerService.	
M2 Template	**M2 Description**
TPS_SYST	This element is used to configure the offer behavior of the server and the find behavior on the client.
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::InitialSdDelayConfig.initialDelayMaxValue	

Mapping Rule	Mapping Type
Take information from SdServerConfig.initialOfferBehavior	full

<table>
<tr><th>BSW Module</th><th colspan="2">BSW Context</th></tr>
<tr><td>Sd</td><td colspan="2">Sd/SdConfig/SdInstance/SdServerTimer</td></tr>
<tr><th colspan="2">BSW Parameter</th><th>BSW Type</th></tr>
<tr><td colspan="2">SdServerTimerInitialOfferDelayMin</td><td>EcucFloatParamDef</td></tr>
<tr><th colspan="3">BSW Description</th></tr>
<tr><td colspan="3">Min value in [s] to delay randomly the first offer. This parameter is mandatory for ServerService.</td></tr>
<tr><th>M2 Template</th><th colspan="2">M2 Description</th></tr>
<tr><td>TPS_SYST</td><td colspan="2">Min Value in seconds to delay radomly the first offer (if aggregated by SdServerConfig) or the transmission of a find message (if aggregated by SdClientConfig).</td></tr>
<tr><th colspan="3">M2 Parameter</th></tr>
<tr><td colspan="3">SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::InitialSdDelayConfig.initialDelayMinValue</td></tr>
<tr><th colspan="2">Mapping Rule</th><th>Mapping Type</th></tr>
<tr><td colspan="2">Take information from SdServerConfig.initialOfferBehavior</td><td>full</td></tr>
</table>

<table>
<tr><th>BSW Module</th><th colspan="2">BSW Context</th></tr>
<tr><td>Sd</td><td colspan="2">Sd/SdConfig/SdInstance/SdServerTimer</td></tr>
<tr><th colspan="2">BSW Parameter</th><th>BSW Type</th></tr>
<tr><td colspan="2">SdServerTimerInitialOfferRepetitionBaseDelay</td><td>EcucFloatParamDef</td></tr>
<tr><th colspan="3">BSW Description</th></tr>
<tr><td colspan="3">The base delay in [s] for offer repetitions. Successive offers have an exponential back off delay (1x base delay, 2x base delay, 4x base delay, ...). This parameter is mandatory for ServerService.</td></tr>
<tr><th>M2 Template</th><th colspan="2">M2 Description</th></tr>
<tr><td>TPS_SYST</td><td colspan="2">The base delay for offer repetitions (if aggregated by SdServerConfig) or find repetitions (if aggregated by SdClientConfig. Successive find messages have an exponential back off delay.</td></tr>
<tr><th colspan="3">M2 Parameter</th></tr>
<tr><td colspan="3">SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::InitialSdDelayConfig.initialRepetitionBaseDelay</td></tr>
<tr><th colspan="2">Mapping Rule</th><th>Mapping Type</th></tr>
<tr><td colspan="2">Take information from SdServerConfig.initialOfferBehavior</td><td>full</td></tr>
</table>

<table>
<tr><th>BSW Module</th><th colspan="2">BSW Context</th></tr>
<tr><td>Sd</td><td colspan="2">Sd/SdConfig/SdInstance/SdServerTimer</td></tr>
<tr><th colspan="2">BSW Parameter</th><th>BSW Type</th></tr>
<tr><td colspan="2">SdServerTimerInitialOfferRepetitionsMax</td><td>EcucIntegerParamDef</td></tr>
<tr><th colspan="3">BSW Description</th></tr>
<tr><td colspan="3">Configure the maximum amount of offer repetition. This parameter is mandatory for ServerService.</td></tr>
<tr><th>M2 Template</th><th colspan="2">M2 Description</th></tr>
<tr><td>TPS_SYST</td><td colspan="2">Describes the maximum amount of offer repetitions (if aggregated by SdServerConfig) or the maximum amount of find repetitions (if aggregated by SdClientConfig).</td></tr>
<tr><th colspan="3">M2 Parameter</th></tr>
<tr><td colspan="3">SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::InitialSdDelayConfig.initialRepetitionsMax</td></tr>
<tr><th colspan="2">Mapping Rule</th><th>Mapping Type</th></tr>
<tr><td colspan="2">Take information from SdServerConfig.initialOfferBehavior</td><td>full</td></tr>
</table>

<table>
<tr><th>BSW Module</th><th colspan="3">BSW Context</th></tr>
<tr><td>Sd</td><td colspan="3">Sd/SdConfig/SdInstance/SdServerTimer</td></tr>
<tr><th colspan="2">BSW Parameter</th><th colspan="2">BSW Type</th></tr>
<tr><td colspan="2">SdServerTimerOfferCyclicDelay</td><td colspan="2">EcucFloatParamDef</td></tr>
<tr><th colspan="4">BSW Description</th></tr>
<tr><td colspan="4">Interval between cyclic offers in the main phase. This parameter is mandatory for ServerService.</td></tr>
<tr><th>M2 Template</th><th colspan="3">M2 Description</th></tr>
<tr><td>TPS_SYST</td><td colspan="3">Optional attribute to define cyclic announcements. Cyclic announcement is active, if the delay is set (in seconds).</td></tr>
<tr><th colspan="4">M2 Parameter</th></tr>
<tr><td colspan="4">SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::SdServerConfig.offerCyclicDelay</td></tr>
<tr><th colspan="3">Mapping Rule</th><th>Mapping Type</th></tr>
<tr><td colspan="3">1:1 mapping</td><td>full</td></tr>
</table>

<table>
<tr><th>BSW Module</th><th colspan="3">BSW Context</th></tr>
<tr><td>Sd</td><td colspan="3">Sd/SdConfig/SdInstance/SdServerTimer</td></tr>
<tr><th colspan="2">BSW Parameter</th><th colspan="2">BSW Type</th></tr>
<tr><td colspan="2">SdServerTimerRequestResponseMaxDelay</td><td colspan="2">EcucFloatParamDef</td></tr>
<tr><th colspan="4">BSW Description</th></tr>
<tr><td colspan="4">Maximum allowable response delay to entries received by multicast in seconds.</td></tr>
<tr><th>M2 Template</th><th colspan="3">M2 Description</th></tr>
<tr><td>TPS_SYST</td><td colspan="3">Maximum allowable response delay to entries received by multicast in seconds.</td></tr>
<tr><th colspan="4">M2 Parameter</th></tr>
<tr><td colspan="4">SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::RequestResponseDelay.maxValue</td></tr>
<tr><th colspan="3">Mapping Rule</th><th>Mapping Type</th></tr>
<tr><td colspan="3">1:1 mapping</td><td>full</td></tr>
</table>

<table>
<tr><th>BSW Module</th><th colspan="3">BSW Context</th></tr>
<tr><td>Sd</td><td colspan="3">Sd/SdConfig/SdInstance/SdServerTimer</td></tr>
<tr><th colspan="2">BSW Parameter</th><th colspan="2">BSW Type</th></tr>
<tr><td colspan="2">SdServerTimerRequestResponseMinDelay</td><td colspan="2">EcucFloatParamDef</td></tr>
<tr><th colspan="4">BSW Description</th></tr>
<tr><td colspan="4">Minimum allowable response delay to entries received by multicast in seconds.</td></tr>
<tr><th>M2 Template</th><th colspan="3">M2 Description</th></tr>
<tr><td>TPS_SYST</td><td colspan="3">Minimum allowable response delay to entries received by multicast in seconds.</td></tr>
<tr><th colspan="4">M2 Parameter</th></tr>
<tr><td colspan="4">SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::RequestResponseDelay.minValue</td></tr>
<tr><th colspan="3">Mapping Rule</th><th>Mapping Type</th></tr>
<tr><td colspan="3">1:1 mapping</td><td>full</td></tr>
</table>

<table>
<tr><th>BSW Module</th><th colspan="3">BSW Context</th></tr>
<tr><td>Sd</td><td colspan="3">Sd/SdConfig/SdInstance/SdServerTimer</td></tr>
<tr><th colspan="2">BSW Parameter</th><th colspan="2">BSW Type</th></tr>
<tr><td colspan="2">SdServerTimerTTL</td><td colspan="2">EcucIntegerParamDef</td></tr>
<tr><th colspan="4">BSW Description</th></tr>
<tr><td colspan="4">Time to live for offer service.</td></tr>
<tr><th>M2 Template</th><th colspan="3">M2 Description</th></tr>
<tr><td>TPS_SYST</td><td colspan="3">Time to live. Shall be a positive value (sInt32).</td></tr>
<tr><th colspan="4">M2 Parameter</th></tr>
<tr><td colspan="4">SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::SdServerConfig.ttl</td></tr>
</table>

Mapping Rule	Mapping Type
1:1 mapping	full

BSW Module	BSW Context		
Sd	Sd		
BSW Parameter		**BSW Type**	
SdGeneral		EcucParamConfContainerDef	
BSW Description			
This container lists the general configuration parameters for the Service Discovery module.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
Sd	Sd/SdGeneral		
BSW Parameter		**BSW Type**	
SdDevErrorDetect		EcucBooleanParamDef	
BSW Description			
Enables and disables the development error detection and notification mechanism.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
Sd	Sd/SdGeneral		
BSW Parameter		**BSW Type**	
SdMainFunctionCycleTime		EcucFloatParamDef	
BSW Description			
This parameter defines the cycle time in seconds of the periodic calling of Sd main function.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context	
Sd	Sd/SdGeneral	
BSW Parameter		**BSW Type**
SdVersionInfoApi		EcucBooleanParamDef
BSW Description		
Enables and disables the version info API.		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

D.6.4 SoAd

BSW Module	**BSW Context**	
SoAd	SoAd	
BSW Parameter		**BSW Type**
SoAdBswModules		EcucParamConfContainerDef
BSW Description		
Each container describes a specific BSW module that the SoAd shall interface to.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
SoAd	SoAd/SoAdBswModules	
BSW Parameter		**BSW Type**
SoAdBswModuleRef		EcucForeignReferenceDef
BSW Description		
This is a reference to one BSW module's configuration (i.e. not the ECUC parameter definition template). Example, there could be several configurations of PduR and this reference selects one of them. SoAd has to figure out from the structure of the referenced BSW module's configuration, what kind of upper layer he deals with. In case of a CDD SoAd expects UL-APIs in form of _SoAd<If|Tp><function> and expects CDD Pdu configuration structures according to the Ecu Configuration specification (chapter CDD module\Socket Adaptor). In case it is one of the standardized AUTOSAR BSW modules, the configuration structures and API names for interaction with SoAd are defined in the corresponding SWS.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
SoAd	SoAd/SoAdBswModules	
BSW Parameter		**BSW Type**
SoAdIf		EcucBooleanParamDef
BSW Description		

Specifies if the BSW module supports the Communication Interface APIs or not. Value true means that the APIs are supported. A module can have both Communication Interface APIs and Transport Protocol APIs (e.g. the PduR module).		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
SoAd	SoAd/SoAdBswModules	
BSW Parameter		**BSW Type**
SoAdIfTriggerTransmit		EcucBooleanParamDef
BSW Description		
Specifies if the BSW module supports the TriggerTransmit API or not. Value true means that the API is supported.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
SoAd	SoAd/SoAdBswModules	
BSW Parameter		**BSW Type**
SoAdIfTxConfirmation		EcucBooleanParamDef
BSW Description		
Specifies if the BSW module supports the TxConfirmation API or not. Value true means that the API is supported.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
SoAd	SoAd/SoAdBswModules	
BSW Parameter		**BSW Type**
SoAdLocalIpAddrAssigmentChg		EcucBooleanParamDef
BSW Description		
Specifies if the BSW module supports the LocalIpAddrAssigmentChg API or not. Value true means that the API is supported.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
SoAd	SoAd/SoAdBswModules	
BSW Parameter		**BSW Type**
SoAdSoConModeChg		EcucBooleanParamDef
BSW Description		
Specifies if the BSW module supports the SoConModeChg API or not. Value true means that the API is supported.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
SoAd	SoAd/SoAdBswModules	
BSW Parameter		**BSW Type**
SoAdTp		EcucBooleanParamDef
BSW Description		
Specifies if the BSW module supports the TransportProtocol APIs or not. Value true means that the APIs are supported. A module can have both Communication Interface APIs and Transport Protocol APIs (e.g. the PduR module).		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
SoAd	SoAd/SoAdBswModules	
BSW Parameter		**BSW Type**
SoAdUseCallerInfix		EcucBooleanParamDef
BSW Description		
Specifies if SoAd shall use (TRUE) the infix "SoAd" when calling an upper layer module function or not (FALSE). E.g. if SoAdUseCallerInfix is TRUE for the upper layer "ABC" then SoAd will call ABC_SoAdIfRxIndication() otherwise SoAd would call ABC_IfRxIndication().		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
SoAd	SoAd/SoAdBswModules	
BSW Parameter		**BSW Type**
SoAdUseTypeInfix		EcucBooleanParamDef
BSW Description		

Specifies if SoAd shall use (TRUE) the API type infix "Tp" or "If" when calling an upper layer module function or not (FALSE). E.g. if SoAdUseTypeInfix is TRUE for the upper layer "ABC" then SoAd will call ABC_IfRxIndication(), otherwise SoAd would call ABC_RxIndication().	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**	
SoAd	SoAd	
BSW Parameter		**BSW Type**
SoAdConfig		EcucParamConfContainerDef
BSW Description		
This container contains the configuration parameters and sub containers of the AUTOSAR SoAd module. This container is a MultipleConfigurationContainer, i.e. this container and its sub-containers exist once per configuration set.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
SoAd	SoAd/SoAdConfig	
BSW Parameter		**BSW Type**
SoAdConfigurationId		EcucIntegerParamDef
BSW Description		
Identification of the SoAd configuration.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
SoAd	SoAd/SoAdConfig	
BSW Parameter		**BSW Type**
SoAdPduRoute		EcucParamConfContainerDef
BSW Description		
Describes the path of a PDU from an upper layer of the SoAd to the socket in the TCP/IP stack for transmission.		
M2 Template	**M2 Description**	
TPS_SYST	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.	
M2 Parameter		

SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication::SocketConnection.pdu	
Mapping Rule	**Mapping Type**
The SocketRoute shall be derived from the from the SocketConnection element and the reference to the Pdu via the SocketConnectionIPduIdentifier.	full

BSW Module	**BSW Context**	
SoAd	SoAd/SoAdConfig/SoAdPduRoute	
BSW Parameter		**BSW Type**
SoAdPduRouteDest		EcucParamConfContainerDef
BSW Description		
Specifies the PDU route destination.		
M2 Template	**M2 Description**	
TPS_SYST	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.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication::SocketConnection.pdu		
Mapping Rule		**Mapping Type**
Container shall be created for every Pdu that is transmitted by the regarded Ecu on a SocketConnection.		full

BSW Module	**BSW Context**	
SoAd	SoAd/SoAdConfig/SoAdPduRoute/SoAdPduRouteDest	
BSW Parameter		**BSW Type**
SoAdTxPduHeaderId		EcucIntegerParamDef
BSW Description		
ID to be sent on the TCP/IP connection if the PDU header option is enabled.		
M2 Template	**M2 Description**	
TPS_SYST	If multiple Pdus are transmitted over the same connection this headerId can be used to distinguish between the different Pdus.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication::SocketConnectionIpduIdentifier.headerId		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
SoAd	SoAd/SoAdConfig/SoAdPduRoute/SoAdPduRouteDest	
BSW Parameter		**BSW Type**
SoAdTxRoutingGroupRef		EcucReferenceDef
BSW Description		
Reference to the routing group.		
M2 Template	**M2 Description**	
TPS_SYST	Reference to the RoutingGroup which can be enabled or disabled.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication::SocketConnectionIpduIdentifier.routingGroup		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdPduRoute/SoAdPduRouteDest	
BSW Parameter		**BSW Type**
SoAdTxSocketConnectionRef		EcucReferenceDef
BSW Description		
Connection on which the PDU is to be sent on, references the appropriate entry in the Socket Connection Table.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdPduRoute/SoAdPduRouteDest	
BSW Parameter		**BSW Type**
SoAdTxUdpTriggerMode		EcucEnumerationParamDef
BSW Description		
Specifies whether a PDU triggers the transmission of the nPduUdpTxBuffer. If this parameter is set to TRIGGER_NEVER, SoAd shall use an nPduUdpTxBuffer for the related socket connection. nPduUdpTxBuffer can only be used for upper layers with IF API, i.e. this parameter shall only be set to TRIGGER_NEVER if all upper layers belonging to the related socket connection have SoAdTxUpperLayerType set to "IF". This parameter is only relevant for UDP connections.		
M2 Template	**M2 Description**	
TPS_SYST	Defines whether the referenced Pdu contributes to the triggering of the socket transmission if Pdu collection is enabled for this socket.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication::SocketConnectionIpduIdentifier.pduCollectionTrigger		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdPduRoute	
BSW Parameter		**BSW Type**
SoAdTxPduId		EcucIntegerParamDef
BSW Description		
Tx PDU ID of the PDU coming from the PDU Router.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdPduRoute	
BSW Parameter		**BSW Type**

SoAdTxPduRef		EcucReferenceDef	
BSW Description			
Reference to the global PDU structure			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
SoAd	SoAd/SoAdConfig/SoAdPduRoute		
BSW Parameter		**BSW Type**	
SoAdTxUpperLayerType		EcucEnumerationParamDef	
BSW Description			
Specifies the upper layer interface type (must be "IF" in case of multiple PduRoutes).			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
SoAd	SoAd/SoAdConfig		
BSW Parameter		**BSW Type**	
SoAdRoutingGroup		EcucParamConfContainerDef	
BSW Description			
Each container describes a specific routing group which can be enabled or disabled. A routing group consists of PDUs. Routing of PDUs can either be forwarding of PDUs from the upper layer to a TCP or UDP socket of the TCP/IP stack specified by a SoAdPduRoute or the other way around specified by a SoAdSocketRoute.			
M2 Template	**M2 Description**		
TPS_SYST	Routing of Pdus in the SoAd can be activated or deactivated. The ShortName of this element shall contain the RoutingGroupId.		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication::SoAdRoutingGroup			
Mapping Rule			**Mapping Type**
Create container for every SoAdRoutingGroup element that is available in the Ecu Extract.			full

BSW Module	**BSW Context**		
SoAd	SoAd/SoAdConfig/SoAdRoutingGroup		
BSW Parameter		**BSW Type**	
SoAdRoutingGroupId		EcucIntegerParamDef	
BSW Description			
Unique ID of Routing Group			
M2 Template	**M2 Description**		
M2 Parameter			

Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdRoutingGroup	
BSW Parameter		**BSW Type**
SoAdRoutingGroupIsEnabledAtInit		EcucBooleanParamDef
BSW Description		
If set to true this routing group will be enabled after initializing the SoAd module (i.e. enabled in the SoAd_Init function).		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdRoutingGroup	
BSW Parameter		**BSW Type**
SoAdRoutingGroupTxTriggerable		EcucBooleanParamDef
BSW Description		
Specifies if the If-TxPDUs related to the PduRouteDest containers referenced by this routing group can be triggered via SoAd_IfRoutingGroupTransmit (TRUE) or not (FALSE).		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig	
BSW Parameter		**BSW Type**
SoAdSocketConnectionGroup		EcucParamConfContainerDef
BSW Description		
Specifies the configuration of a socket connection group, i.e. specifies the socket connections belonging to the group and the parameters which are common for all socket connections of the group. A socket connection specifies how data can be received and transmited via a TCP or UDP socket.		
M2 Template	**M2 Description**	
TPS_SYST	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.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication::SocketConnectionBundle		
Mapping Rule		**Mapping Type**
SocketConnectionGroups shall be derived from SocketConnectionBundles. For every existing SocketConnectionBundle a SocketConnectionGroup shall be created.		full

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup	
BSW Parameter		**BSW Type**
SoAdPduHeaderEnable		EcucBooleanParamDef
BSW Description		
Enables the transmission of the PDU header (ID, length) on this socket connection. TRUE: add SoAd PDU header before PDU data FALSE: No SoAd PDU header is used		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup	
BSW Parameter		**BSW Type**
SoAdResourceManagementEnable		EcucBooleanParamDef
BSW Description		
Enables the resource management option for this socket. May not be activated for UDP sockets in receive. TRUE: resource management option enabled FALSE: resource management option disabled		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup	
BSW Parameter		**BSW Type**
SoAdSocketAutomaticSoConSetup		EcucBooleanParamDef
BSW Description		
Specifies if the setup of the socket connection shall be done automatically (TRUE) or manually (FALSE) via SoAd_OpenSoCon() and SoAd_CloseSoCon().		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup	
BSW Parameter		**BSW Type**

SoAdSocketConnection		EcucParamConfContainerDef	
BSW Description			
Specifies the socket connection (Id and remote address information). Note: Parameters which are common to all socket connections of a socket connection group are specified directly at the group.			
M2 Template	**M2 Description**		
TPS_SYST	Remote Port for TCP/UDP connection. May be different for each Frame or use the same remote port. In second case headerId attribute needs to be considered.		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication::SocketConnection			
Mapping Rule			**Mapping Type**
SocketConnections shall be derived from SocketConnection elements that are aggregated by the SocketConnectionBundle. For every existing SocketConnection in the SystemTemplate a SocketConnection in the SoAd Config shall be created.			full

BSW Module	**BSW Context**		
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketConnection		
BSW Parameter		**BSW Type**	
SoAdSocketId		EcucIntegerParamDef	
BSW Description			
Socket connection identifier used as SoConId in the interaction with upper layers.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketConnection		
BSW Parameter		**BSW Type**	
SoAdSocketRemoteAddress		EcucParamConfContainerDef	
BSW Description			
Subcontainer of SoAdSocketConnection to specify the remote address (IP address and port) for a socket connection. If SoAdSocketRemoteAddress is not specified the remote address has to be set by the upper layer via SoAd_SetRemoteAddr().			
M2 Template	**M2 Description**		
TPS_SYST	Client Port for TCP/UDP connection.		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication::SocketConnection.clientPort			
Mapping Rule			**Mapping Type**
The remoteAddress shall be derived from the aggregated ApplicationEndpoint and the NetworkEndpoint			full

BSW Module	**BSW Context**		
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketConnection/SoAdSocketRemoteAddress		
BSW Parameter		**BSW Type**	

SoAdSocketRemoteIpAddress		EcucStringParamDef	
BSW Description			
IP address of remote node. To accept any remote IP address, set SoAdSocketRemoteIpAddress to "ANY". See message acceptance policy for more details.			
M2 Template	**M2 Description**		
TPS_SYST	To build a valid network endpoint address there has to be either one MAC multicast group reference or an ipv4 configuration or an ipv6 configuration.		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::NetworkEndpointAddress			
Mapping Rule			**Mapping Type**
Shall be derived from NetworkEndpointAddress at the NetworkEndpoint.			full

BSW Module	**BSW Context**		
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketConnection/SoAdSocketRemoteAddress		
BSW Parameter		**BSW Type**	
SoAdSocketRemotePort		EcucIntegerParamDef	
BSW Description			
Remote UDP or TCP port used for this connection. To accept any remote port, set SoAdSocketRemotePort to 0. See message acceptance policy for more details.			
M2 Template	**M2 Description**		
TPS_SYST	Definition of a Port Number.		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::TpPort.portNumber			
Mapping Rule			**Mapping Type**
Shall be derived from TpConfiguration at the ApplicationEndpoint.			full

BSW Module	**BSW Context**		
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup		
BSW Parameter		**BSW Type**	
SoAdSocketFramePriority		EcucIntegerParamDef	
BSW Description			
Specifies the priority of the Ethernet frame. If IEEE 802.1Q VLAN Tags are used, the specified priority will be used in the VLAN Tag PCP filed. If this optional parameter is not available the default priority specified in the TcpIp module is used.			
M2 Template	**M2 Description**		
TPS_SYST	It shall be considered that the priority in the SoAd is defined only once per SocketConnectionGroup. The SocketConnections in the system description shall be created adequate.		
M2 Parameter			
Mapping Rule			**Mapping Type**
The priority in the system description can be defined at the ProvidedServiceInstance, at the NetworkEndpoint or the ApplicationEndpoint.			full

BSW Module	**BSW Context**	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup	
BSW Parameter		**BSW Type**
SoAdSocketIpAddrAssignmentChgNotification		EcucBooleanParamDef

BSW Description	
Specifies if the local IP address assignment change notification callback function of the upper layer shall be called if the assignment of the local IP address used by this socket connection changes.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup
BSW Parameter	**BSW Type**
SoAdSocketLocalAddressRef	EcucSymbolicNameReferenceDef
BSW Description	
Local IP address and interface used for this connection.	
M2 Template	**M2 Description**
TPS_SYST	Server Port for TCP/UDP connection.
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication::SocketConnectionBundle.serverPort	
Mapping Rule	**Mapping Type**
The IP Address shall be derived from the NetworkEndpoint that referenced by the ApplicationEndpoint that is aggregated by the SocketAddress referenced from the SocketConnectionBundle.	full

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup
BSW Parameter	**BSW Type**
SoAdSocketLocalPort	EcucIntegerParamDef
BSW Description	
Local UDP or TCP port used for this connection.If this parameter set to 0 SoAd requests TcpIp to select an ephemeral port.	
M2 Template	**M2 Description**
TPS_SYST	Server Port for TCP/UDP connection.
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication::SocketConnectionBundle.serverPort	
Mapping Rule	**Mapping Type**
The Port Address shall be derived from the ApplicationEndpoint that is aggregated by the SocketAddress	full

BSW Module	BSW Context
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup
BSW Parameter	**BSW Type**
SoAdSocketMsgAcceptanceFilterEnabled	EcucBooleanParamDef
BSW Description	

Specifies if the message acceptance filter is enabled (TRUE) or not (FALSE). Note: if a wildcard is used in SoAdSocketRemoteAddress AND SoAdSocketUdpListenOnly is FALSE, this parameter must be TRUE. Note: if multiple SoAdSocketConnections are configured for one SoAdSocketConnectionGroup, this parameter must be TRUE.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup
BSW Parameter	**BSW Type**
SoAdSocketProtocol	EcucChoiceContainerDef
BSW Description	
Specifies the transport protocol and transport protocol specific parameters used for the socket connections of the socket connection group.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol
BSW Parameter	**BSW Type**
SoAdSocketTcp	EcucParamConfContainerDef
BSW Description	
Specifies that TCP is used as transport protocol for the socket connection group and parameters only related to TCP socket connections.	
M2 Template	**M2 Description**
SystemTemplate	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.
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::ApplicationEndpoint	
Mapping Rule	**Mapping Type**
Information shall be derived from the ApplicationEndpoint (TpConfiguration) that is referenced via the serverPort (SocketAddress) by the SocketConnectionBundle.	full

BSW Module	**BSW Context**
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketTcp
BSW Parameter	**BSW Type**
SoAdSocketTcpImmediateTpTxConfirmation	EcucBooleanParamDef
BSW Description	

If set to FALSE, SoAd notifies the TP upper layer via transmit confirmation after a Tcp Ack has been received. If set to TRUE, SoAd notifies the TP upper layer via transmit confirmation immediately after transmit has been accepted by TcpIp.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketTcp	
BSW Parameter		**BSW Type**
SoAdSocketTcpInitiate		EcucBooleanParamDef
BSW Description		
Specifies the initiator for this TCP connection. It will not be defined for UDP sockets. TRUE: This TCP connection is initiated by this module. FALSE: This TCP connection is to be initiated in the listen mode.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketTcp	
BSW Parameter		**BSW Type**
SoAdSocketTcpKeepAlive		EcucBooleanParamDef
BSW Description		
Specifies to use the keep-alive mechanism for this connection. It will not be defined for UDP sockets. TRUE: This TCP connection will use the keep-alive mechanism. FALSE: This TCP connection will not use the keep-alive mechanism. Note: This parameter must not be set to TRUE if TcpIpTcpKeepAliveEnabled is set to FALSE.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketTcp	
BSW Parameter		**BSW Type**
SoAdSocketTcpNoDelay		EcucBooleanParamDef
BSW Description		

Specifies not to use the congestion control mechanism for this connection. It will not be defined for UDP sockets. TRUE: This TCP connection will NOT use congestion control. FALSE: This TCP connection will use congestion control. If the optional parameter is not enabled, the default behavior configured for TcpIp via the parameter TcpIpTcpNagleEnabled is applied. Note: This parameter must not be set to FALSE if TcpIpTcpNagleEnabled is set to FALSE.		
M2 Template	**M2 Description**	
TPS_SYST	Indicates if Nagle's Algorithm is used.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::TcpTp.naglesAlgorithm		
Mapping Rule		**Mapping Type**
If naglesAlgorithm in the EcuExtract is true set this parameter to false. If nagles Algorithm in the EcuExtract is false set this parameter to true. If naglesAlgorithm in the EcuExtract is not defined then this parameter shall not be set.		full

BSW Module	**BSW Context**	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAd SocketTcp	
BSW Parameter		**BSW Type**
SoAdSocketTcpTxQuota		EcucIntegerParamDef
BSW Description		
Specifies the maximum amout of bytes (PDU data provided by the upper layer and PDU Header if used) the SoAd may queue for transmission via TCP at the TcpIp module for each socket connection of this socket connection group. Rationale: prohibits that a socket connection consumes all available transmit buffers at the TcpIp and blocks transmissions via other socket connections. If the optional parameter is not enabled, the amout of data is not limited.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol	
BSW Parameter		**BSW Type**
SoAdSocketUdp		EcucParamConfContainerDef
BSW Description		
Specifies that UDP is used as transport protocol for the socket connection group and parameters only related to UDP socket connections.		
M2 Template	**M2 Description**	
SystemTemplate	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.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::ApplicationEndpoint		
Mapping Rule		**Mapping Type**

Information shall be derived from the ApplicationEndpoint (TpConfiguration) that is referenced via the serverPort (SocketAddress) by the SocketConnectionBundle.	full

BSW Module	**BSW Context**	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketUdp	
BSW Parameter		**BSW Type**
SoAdSocketUdpListenOnly		EcucBooleanParamDef
BSW Description		
Specifies if the socket connection group is only used for reception (TRUE) or used for both reception and transmission (FALSE).		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketUdp	
BSW Parameter		**BSW Type**
SoAdSocketUdpRetryEnabled		EcucBooleanParamDef
BSW Description		
Specifies if an UdpTxBuffer shall be used (TRUE) to retry an UDP transmission in case of TCPIP_E_ARP_CACHE_MISS or not (FALSE).		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAdSocketUdp	
BSW Parameter		**BSW Type**
SoAdSocketUdpTriggerTimeout		EcucFloatParamDef
BSW Description		
Specifies the timeout in [s] a nPduUdpTxBuffer is waiting for a PDU with TriggerMode = TRIGGER_ALWAYS, i.e. when the timeout expires the nPduUdpTxBuffer is transmitted. Timer is reset after each UDP transmission. This optional parameter is only relevant if a nPduUdpTxBuffer is used.		
M2 Template	**M2 Description**	
TPS_SYST	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.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication::SocketConnection.pduCollectionTimeout		
Mapping Rule		**Mapping Type**

1:1 mapping	full

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup/SoAdSocketProtocol/SoAd SocketUdp	
BSW Parameter		**BSW Type**
SoAdSocketnPduUdpTxBufferMin		EcucIntegerParamDef
BSW Description		
Specifies the amout of data in bytes (PDU data provided by the upper layer and PDU Header if used) the SoAd shall be able to buffer for data transmission via this socket connection in case the UDP message shall be buffered for transmission of multiple PDUs per UDP. Note: in case of a UDP socket and an upper layer with TP API or an upper layer with IF API with UDP transmit retry (for single PDUs) configured, the required buffer size can be determined automatically. This optional parameter is only relevant if a nPduUdpTxBuffer is used.		
M2 Template	**M2 Description**	
TPS_SYST	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.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication::SocketConnection.pduCollectionMaxBufferSize		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup	
BSW Parameter		**BSW Type**
SoAdSocketSoConModeChgNotification		EcucBooleanParamDef
BSW Description		
Specifies if the SoCon mode change notification callback function of the upper layer shall be called in case of SoCon mode change.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketConnectionGroup	
BSW Parameter		**BSW Type**
SoAdSocketTpRxBufferMin		EcucIntegerParamDef
BSW Description		
Specifies the amout of data in bytes (PDU data for the upper layer and PDU Header if used) the SoAd shall at least be able to buffer for data reception via each socket connection of the socket connection group and using an upper layer with TP. Note: in case of a TCP socket where PduHeaderMode is used and an upper layer with IF-API, the required buffer size can be determined automatically.		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig	
BSW Parameter		**BSW Type**
SoAdSocketRoute		EcucParamConfContainerDef
BSW Description		
Describes the path of a PDU from a socket in the TCP/IP stack to an upper layer of the SoAd after reception in the TCP/IP Stack.		
M2 Template	**M2 Description**	
TPS_SYST	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.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication::SocketConnection.pdu		
Mapping Rule		**Mapping Type**
The SocketRoute shall be derived from the from the SocketConnection element and the reference to the Pdu via the SocketConnectionIPduIdentifier.		full

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketRoute	
BSW Parameter		**BSW Type**
SoAdRxPduHeaderId		EcucIntegerParamDef
BSW Description		
ID contained in the packet received on the TCP/IP connection if the PDU header option is enabled.		
M2 Template	**M2 Description**	
TPS_SYST	If multiple Pdus are transmitted over the same connection this headerId can be used to distinguish between the different Pdus.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication::SocketConnectionIpduIdentifier.headerId		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketRoute	
BSW Parameter		**BSW Type**
SoAdRxSocketConnectionRef		EcucReferenceDef
BSW Description		
Connection on which the PDU was received. This references an entry in the Socket Connection Table.		
M2 Template	**M2 Description**	
TPS_SYST	The SoAd serves as a (De)Multiplexer between different PDU sources and the TCP/IP stack.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication::SocketConnection		
Mapping Rule		**Mapping Type**

In the EcuExtract the SocketConnectionIpduIdentifier is aggregated by the SocketConnection in the role "pdu".	full

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketRoute	
BSW Parameter		**BSW Type**
SoAdSocketRouteDest		EcucParamConfContainerDef
BSW Description		
Describes the upper layer destination PDU for a message received on a TcpIp socket.		
M2 Template	**M2 Description**	
TPS_SYST	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.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication::SocketConnection.pdu		
Mapping Rule		**Mapping Type**
Container shall be created for every Pdu that is received by the regarded ECU on the SocketConnection.		full

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketRoute/SoAdSocketRouteDest	
BSW Parameter		**BSW Type**
SoAdRxPduId		EcucIntegerParamDef
BSW Description		
This unique identifier is used for a receive cancellation request from an upper layer of the SoAd.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketRoute/SoAdSocketRouteDest	
BSW Parameter		**BSW Type**
SoAdRxPduRef		EcucReferenceDef
BSW Description		
Reference to the global PDU structure		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
SoAd	SoAd/SoAdConfig/SoAdSocketRoute/SoAdSocketRouteDest	
BSW Parameter		**BSW Type**
SoAdRxRoutingGroupRef		EcucReferenceDef

BSW Description		
Reference to the routing group.		
M2 Template	**M2 Description**	
TPS_SYST	Reference to the RoutingGroup which can be enabled or disabled.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication::SocketConnectionIpduIdentifier.routingGroup		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
SoAd	SoAd/SoAdConfig/SoAdSocketRoute/SoAdSocketRouteDest	
BSW Parameter		**BSW Type**
SoAdRxUpperLayerType		EcucEnumerationParamDef
BSW Description		
Specifies the upper layer interface type (must be "IF" in case of multiple RxPdus).		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
SoAd	SoAd	
BSW Parameter		**BSW Type**
SoAdGeneral		EcucParamConfContainerDef
BSW Description		
This container contains all global configuration parameters of SoAd.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
SoAd	SoAd/SoAdGeneral	
BSW Parameter		**BSW Type**
SoAdDevErrorDetect		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling development error detection support.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context		
SoAd	SoAd/SoAdGeneral		
BSW Parameter		**BSW Type**	
SoAdIPv6AddressEnabled		EcucBooleanParamDef	
BSW Description			
Allows for increased memory allocation to store IPv6 addresses. TRUE: Enables support for IPv6 addresses FALSE: Only IPv4 addresses are supported			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
SoAd	SoAd/SoAdGeneral		
BSW Parameter		**BSW Type**	
SoAdMainFunctionPeriod		EcucFloatParamDef	
BSW Description			
Determines the frequency at which the SoAd_MainFunction() is called in [s].			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
SoAd	SoAd/SoAdGeneral		
BSW Parameter		**BSW Type**	
SoAdRoutingGroupMax		EcucIntegerParamDef	
BSW Description			
Specifies the maximum number of SoAd routing groups. Furthermore it defines the platform type used for RoutingGroupIdType. If SoAdRoutingGroupMax is not greater than 256, an uint8 is used, otherwise an uint16.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context	
SoAd	SoAd/SoAdGeneral	
BSW Parameter		**BSW Type**
SoAdSoConMax		EcucIntegerParamDef
BSW Description		

Specifies the maximum number of SoAd socket connections. Furthermore it defines the platform type used for SoAd_SoConIdType. If SoAdSoConMax is not greater than 256, an uint8 is used, otherwise uint16.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**		
SoAd	SoAd/SoAdGeneral		
BSW Parameter		**BSW Type**	
SoAdVersionInfoApi		EcucBooleanParamDef	
BSW Description			
Activates the SoAd_GetVersionInfo() API. TRUE: Enables the SoAd_GetVersionInfo() API. FALSE: SoAd_GetVersionInfo() API is not included.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

//

D.6.5 EthSM

// //

BSW Module	**BSW Context**		
EthSM	EthSM		
BSW Parameter		**BSW Type**	
EthSMGeneral		EcucParamConfContainerDef	
BSW Description			
This container contains the global parameter of the Ethernet State Manager.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**	
EthSM	EthSM/EthSMGeneral	
BSW Parameter		**BSW Type**
EthSMDevErrorDetect		EcucBooleanParamDef
BSW Description		

Enables and disables the development error detection and notification mechanism.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
EthSM	EthSM/EthSMGeneral	
BSW Parameter		**BSW Type**
EthSMDummyMode		EcucBooleanParamDef
BSW Description		
Disables the API to the EthIf. The API to the ComM is available but the functionality is deactivated. The function calls from the ComM will be answered with the return value E_OK.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
EthSM	EthSM/EthSMGeneral	
BSW Parameter		**BSW Type**
EthSMMainFunctionPeriod		EcucFloatParamDef
BSW Description		
Specifies the period in seconds that the MainFunction has to be triggered with.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
EthSM	EthSM/EthSMGeneral	
BSW Parameter		**BSW Type**
EthSMVersionInfoApi		EcucBooleanParamDef
BSW Description		
Enables and disables the version info API.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**

EthSM	EthSM
BSW Parameter	**BSW Type**
EthSMNetwork	EcucParamConfContainerDef
BSW Description	
This container contains the Ethernet network-specific parameters of each Ethernet network. It also contains the controller and transceiver IDs assigned to a Ethernet network.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
EthSM	EthSM/EthSMNetwork
BSW Parameter	**BSW Type**
EthSMComMNetworkHandleRef	EcucSymbolicNameReferenceDef
BSW Description	
Unique handle to identify one certain Ethernet network. Reference to one of the network handles configured for the ComM.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
EthSM	EthSM/EthSMNetwork
BSW Parameter	**BSW Type**
EthSMConfirmationTimeout	EcucFloatParamDef
BSW Description	
Please note that this parameter is deprecated and will be removed in future. Timeout in seconds for the calls to EthIf: EthIf_ControllerInit EthIf_TransceiverInit EthIf_SetControllerMode EthIf_SetTransceiverMode	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
EthSM	EthSM/EthSMNetwork
BSW Parameter	**BSW Type**
EthSMDemEventParameterRefs	EcucParamConfContainerDef

BSW Description		
Container for the references to DemEventParameter elements which shall be invoked using the API Dem_ReportErrorStatus API in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId value. The standardized errors are provided in the container and can be extended by vendor specific error references.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
EthSM	EthSM/EthSMNetwork/EthSMDemEventParameterRefs	
BSW Parameter		**BSW Type**
ETHSM_E_LINK_DOWN		EcucSymbolicNameReferenceDef
BSW Description		
Reference to configured DEM event to report bus off errors for this Eth network.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
EthSM	EthSM/EthSMNetwork	
BSW Parameter		**BSW Type**
EthSMEthIfControllerRef		EcucSymbolicNameReferenceDef
BSW Description		
Reference to EthIfCtrl container where a ETH controller and transceiver (optional) combination is configured.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

//

D.6.6 EthTrcv

// //

BSW Module	BSW Context	
EthTrcv	EthTrcv	
BSW Parameter		**BSW Type**
EthTrcvConfigSet		EcucParamConfContainerDef

BSW Description		
All underlying parameters may be part of a multiple configuration set.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context		
EthTrcv	EthTrcv/EthTrcvConfigSet		
BSW Parameter		**BSW Type**	
EthTrcvConfig		EcucParamConfContainerDef	
BSW Description			
Configuration of the individual transceiver			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig		
BSW Parameter		**BSW Type**	
EthTrcvAutoNegotiationEnabled		EcucBooleanParamDef	
BSW Description			
Specifies if Auto-Negotiation is enabled (TRUE) or disabled (FALSE) for determination of the Ethernet transceiver speed.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig		
BSW Parameter		**BSW Type**	
EthTrcvCtrlIdx		EcucIntegerParamDef	
BSW Description			
Specifies the controller used for MII access to the transceiver			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig		
BSW Parameter		**BSW Type**	
EthTrcvDemEventParameterRefs		EcucParamConfContainerDef	
BSW Description			
Container for the references to DemEventParameter elements which shall be invoked using the API Dem_ReportErrorStatus API in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId value. The standardized errors are provided in the container and can be extended by vendor specific error references.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig/EthTrcvDemEventParameterRefs		
BSW Parameter		**BSW Type**	
ETHTRCV_E_ACCESS		EcucSymbolicNameReferenceDef	
BSW Description			
Reference to the DemEventParameter which shall be issued when the error "Transceiver access failed" has occurred.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	BSW Context		
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig		
BSW Parameter		**BSW Type**	
EthTrcvDuplexMode		EcucEnumerationParamDef	
BSW Description			
Specifies the duplex mode of the Ethernet transceiver link if Auto-Negotiation is disabled. This parameter is ignored if Auto-Negotiation is enabled.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context	
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig	
BSW Parameter		**BSW Type**
EthTrcvIdx		EcucIntegerParamDef
BSW Description		
Specifies the instance ID of the configured transceiver.		

M2 Template	M2 Description	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig	
BSW Parameter		**BSW Type**
EthTrcvMiiIdx		EcucIntegerParamDef
BSW Description		
Specifies the transceiver index used for MII access to the transceiver		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
EthTrcv	EthTrcv/EthTrcvConfigSet/EthTrcvConfig	
BSW Parameter		**BSW Type**
EthTrcvSpeed		EcucEnumerationParamDef
BSW Description		
Specifies the speed of the Ethernet transceiver link in [MBit/s]. If AutoNegotiation is enabled this is the maximum speed advertised for Auto-Negotiation.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
EthTrcv	EthTrcv	
BSW Parameter		**BSW Type**
EthTrcvGeneral		EcucParamConfContainerDef
BSW Description		
General configuration of Ethernet Transceiver Driver module		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context
EthTrcv	EthTrcv/EthTrcvGeneral

BSW Parameter			BSW Type	
EthTrcvDevErrorDetect			EcucBooleanParamDef	
BSW Description				
Enables / Disables development error detection				
M2 Template	**M2 Description**			
M2 Parameter				
Mapping Rule				**Mapping Type**
				local

BSW Module	BSW Context			
EthTrcv	EthTrcv/EthTrcvGeneral			
BSW Parameter			**BSW Type**	
EthTrcvGetBaudRateApi			EcucBooleanParamDef	
BSW Description				
Enables / Disables EthTrcv_GetBaudRate API				
M2 Template	**M2 Description**			
M2 Parameter				
Mapping Rule				**Mapping Type**
				local

BSW Module	BSW Context			
EthTrcv	EthTrcv/EthTrcvGeneral			
BSW Parameter			**BSW Type**	
EthTrcvGetDuplexModeApi			EcucBooleanParamDef	
BSW Description				
Enables / Disables EthTrcv_GetDuplexMode API				
M2 Template	**M2 Description**			
M2 Parameter				
Mapping Rule				**Mapping Type**
				local

BSW Module	BSW Context			
EthTrcv	EthTrcv/EthTrcvGeneral			
BSW Parameter			**BSW Type**	
EthTrcvGetLinkStateApi			EcucBooleanParamDef	
BSW Description				
Enables / Disables EthTrcv_GetLinkState API				
M2 Template	**M2 Description**			
M2 Parameter				
Mapping Rule				**Mapping Type**
				local

BSW Module	BSW Context	
EthTrcv	EthTrcv/EthTrcvGeneral	
BSW Parameter		**BSW Type**
EthTrcvGetTransceiverModeApi		EcucBooleanParamDef
BSW Description		
Enables / Disables EthTrcv_GetTransceiverMode API		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
EthTrcv	EthTrcv/EthTrcvGeneral	
BSW Parameter		**BSW Type**
EthTrcvIndex		EcucIntegerParamDef
BSW Description		
Specifies the InstanceId of this module instance. If only one instance is present it shall have the Id 0.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
EthTrcv	EthTrcv/EthTrcvGeneral	
BSW Parameter		**BSW Type**
EthTrcvMaxTrcvsSupported		EcucIntegerParamDef
BSW Description		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
EthTrcv	EthTrcv/EthTrcvGeneral	
BSW Parameter		**BSW Type**
EthTrcvSetTransceiverModeApi		EcucBooleanParamDef
BSW Description		
Enables / Disables EthTrcv_SetTransceiverMode API		
M2 Template	**M2 Description**	
M2 Parameter		

Mapping Rule	Mapping Type
	local

BSW Module	BSW Context	
EthTrcv	EthTrcv/EthTrcvGeneral	
BSW Parameter		**BSW Type**
EthTrcvStartAutoNegotiationApi		EcucBooleanParamDef
BSW Description		
Enables / Disables EthTrcv_StartAutoNegotiation API		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
EthTrcv	EthTrcv/EthTrcvGeneral	
BSW Parameter		**BSW Type**
EthTrcvVersionInfoApi		EcucBooleanParamDef
BSW Description		
Enables / Disables version info API		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
EthTrcv	EthTrcv/EthTrcvGeneral	
BSW Parameter		**BSW Type**
EthTrcvVersionInfoApiMacro		EcucBooleanParamDef
BSW Description		
Enables / Disables version info API macro implementation		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

D.6.7 TcpIp

BSW Module	BSW Context	
TcpIp	TcpIp	
BSW Parameter		**BSW Type**

TcpIpConfig		EcucParamConfContainerDef	
BSW Description			
This container contains the configuration parameters and sub containers of the AUTOSAR TcpIp module. This container is a MultipleConfigurationContainer, i.e. this container and its sub-containers exist once per configuration set.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
TcpIp	TcpIp/TcpIpConfig		
BSW Parameter		**BSW Type**	
TcpConfigurationId		EcucIntegerParamDef	
BSW Description			
Identification of the TcpIp configuration.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
TcpIp	TcpIp/TcpIpConfig		
BSW Parameter		**BSW Type**	
TcpIpArpConfig		EcucParamConfContainerDef	
BSW Description			
Specifies the configuration parameters of the ARP (Address Resolution Protocol) sub-module.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
TcpIp	TcpIp/TcpIpConfig/TcpIpArpConfig		
BSW Parameter		**BSW Type**	
TcpIpArpNumGratuitousARPonStartup		EcucIntegerParamDef	
BSW Description			
Specifies the number of gratuitous ARP replies which shall be sent on assignment of a new IP address.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

	local

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpArpConfig	
BSW Parameter		**BSW Type**
TcpIpArpTableEntryTimeout		EcucFloatParamDef
BSW Description		
Timeout in seconds after which an unused ARP entry is removed.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpArpConfig	
BSW Parameter		**BSW Type**
TcpIpArpTableSizeMax		EcucIntegerParamDef
BSW Description		
Maximum number of entries in the ARP table.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig	
BSW Parameter		**BSW Type**
TcpIpAutoIpConfig		EcucParamConfContainerDef
BSW Description		
Specifies the configuration parameters of the Auto-IP (automatic private IP addressing) sub-module.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpAutoIpConfig	
BSW Parameter		**BSW Type**
TcpIpAutoIpInitTimeout		EcucFloatParamDef
BSW Description		
The time in seconds Auto-IP waits at startup, before beginning with ARP probing. This delay is used to give DHCP time to acquire a lease in case a DHCP server is present.		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**	
TcpIp	TcpIp/TcpIpConfig	
BSW Parameter		**BSW Type**
TcpIpCtrl		EcucParamConfContainerDef
BSW Description		
Specifies the EthIf controller used for IP communication and TcpIp errors that shall be reported to DEM.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
TcpIp	TcpIp/TcpIpConfig/TcpIpCtrl	
BSW Parameter		**BSW Type**
TcpIpCtrlDemEventParameterRefs		EcucParamConfContainerDef
BSW Description		
This container is a subcontainer of TcpIpCtrl and specifies the references to DemEventParameter elements which shall be invoked using the API Dem_ReportErrorStatus API in case the corresponding TcpIp error occurs for communication on the EthIf Controller. The EventId is taken from the referenced DemEventParameter's DemEventId value. The standardized errors are provided in the container and can be extended by vendor specific error references.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
TcpIp	TcpIp/TcpIpConfig/TcpIpCtrl/TcpIpCtrlDemEventParameterRefs	
BSW Parameter		**BSW Type**
TCPIP_E_CONNABORTED		EcucSymbolicNameReferenceDef
BSW Description		
Reference to the DemEventParameter which shall be issued when the error "Connection aborted by TcpIp stack because of an error" has occurred.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpCtrl/TcpIpCtrlDemEventParameterRefs	
BSW Parameter		**BSW Type**
TCPIP_E_CONNREFUSED		EcucSymbolicNameReferenceDef
BSW Description		
Reference to the DemEventParameter which shall be issued when the error "Connection refused" has occurred.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpCtrl/TcpIpCtrlDemEventParameterRefs	
BSW Parameter		**BSW Type**
TCPIP_E_CONNRESET		EcucSymbolicNameReferenceDef
BSW Description		
Reference to the DemEventParameter which shall be issued when the error "Connection reset by peer" has occurred.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpCtrl/TcpIpCtrlDemEventParameterRefs	
BSW Parameter		**BSW Type**
TCPIP_E_HOSTDOWN		EcucSymbolicNameReferenceDef
BSW Description		
Reference to the DemEventParameter which shall be issued when the error "Host is down" has occurred.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpCtrl/TcpIpCtrlDemEventParameterRefs	
BSW Parameter		**BSW Type**
TCPIP_E_HOSTUNREACH		EcucSymbolicNameReferenceDef
BSW Description		
Reference to the DemEventParameter which shall be issued when the error "No route to host" has occurred.		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpCtrl/TcpIpCtrlDemEventParameterRefs
BSW Parameter	**BSW Type**
TCPIP_E_NETDOWN	EcucSymbolicNameReferenceDef
BSW Description	
Reference to the DemEventParameter which shall be issued when the error "Network is down" has occurred.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpCtrl/TcpIpCtrlDemEventParameterRefs
BSW Parameter	**BSW Type**
TCPIP_E_NETRESET	EcucSymbolicNameReferenceDef
BSW Description	
Reference to the DemEventParameter which shall be issued when the error "Network dropped connection on reset" has occurred.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpCtrl/TcpIpCtrlDemEventParameterRefs
BSW Parameter	**BSW Type**
TCPIP_E_NETUNREACH	EcucSymbolicNameReferenceDef
BSW Description	
Reference to the DemEventParameter which shall be issued when the error "Network is unreachable" has occurred.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**

TcpIp	TcpIp/TcpIpConfig/TcpIpCtrl/TcpIpCtrlDemEventParameterRefs		
BSW Parameter		**BSW Type**	
TCPIP_E_TIMEDOUT		EcucSymbolicNameReferenceDef	
BSW Description			
Reference to the DemEventParameter which shall be issued when the error "Operation timed out" has occurred.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
TcpIp	TcpIp/TcpIpConfig/TcpIpCtrl		
BSW Parameter		**BSW Type**	
TcpIpEthIfCtrlRef		EcucSymbolicNameReferenceDef	
BSW Description			
Reference to EthIf controller where the IP address shall be assigned.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
TcpIp	TcpIp/TcpIpConfig/TcpIpCtrl		
BSW Parameter		**BSW Type**	
TcpIpIpFramePrioDefault		EcucIntegerParamDef	
BSW Description			
Specifies the default value for the frame priority used by all sockets. Note: the value can be changed for each socket individually via TcpIp_ChangeParameter() service. If this optional parameter is not available, 0 is used as default priority.			
M2 Template	**M2 Description**		
TPS_SYST	Standard output-priority outgoing Frames will be tagged with. This allows to assign different defaultPriorities to each VLAN.		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::VlanMembership.defaultPriority			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	**BSW Context**		
TcpIp	TcpIp/TcpIpConfig		
BSW Parameter		**BSW Type**	
TcpIpIcmpConfig		EcucParamConfContainerDef	
BSW Description			
Specifies the configuration parameters of the ICMP (Internet Control Message Protocol) sub-module.			
M2 Template	**M2 Description**		

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpIcmpConfig	
BSW Parameter		**BSW Type**
TcpIpIcmpMsgHandler		EcucParamConfContainerDef
BSW Description		
This container is a subcontainer of TcpIpIcmpConfig and specifies the configuration parameters for the ICMP message handler.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpIcmpConfig/TcpIpIcmpMsgHandler	
BSW Parameter		**BSW Type**
TcpIpIcmpMsgHandlerHeaderFileName		EcucStringParamDef
BSW Description		
This parameter specifies the name of the header file containing the definition of the ICMP message handler function.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpIcmpConfig/TcpIpIcmpMsgHandler	
BSW Parameter		**BSW Type**
TcpIpIcmpMsgHandlerName		EcucFunctionNameDef
BSW Description		
This parameter defines the name of the ICMP message handler function <User_IcmpMsgHandler>.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpIcmpConfig	
BSW Parameter		**BSW Type**

TcpIpIcmpTtl		EcucIntegerParamDef	
BSW Description			
Default Time-to-live value of outgoing ICMP packets.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
TcpIp	TcpIp/TcpIpConfig		
BSW Parameter		**BSW Type**	
TcpIpIpConfig		EcucParamConfContainerDef	
BSW Description			
Specifies the configuration parameters of the IP (Internet Protocol) sub-module			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig		
BSW Parameter		**BSW Type**	
TcpIpIpFragmentationRxEnabled		EcucBooleanParamDef	
BSW Description			
Enables (TRUE) or disables (FALSE) support for reassembling of incoming datagrams that are fragmented according to IETF RFC 815 (IP Datagram Reassembly Algorithms).			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig		
BSW Parameter		**BSW Type**	
TcpIpIpNumFragments		EcucIntegerParamDef	
BSW Description			
Specifies the maximum number of IP fragments per datagram. Note: this parameter is only relevant if TcpIpIpFragmentationRxEnabled is TRUE.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**	
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig	
BSW Parameter		**BSW Type**
TcpIpIpNumReassDgrams		EcucIntegerParamDef
BSW Description		
Specifies the maximum number of fragmented IP datagrams that can be reassembled in parallel. Note: this parameter is only relevant if TcpIpIpFragmentationRxEnabled is TRUE.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
TcpIp	TcpIp/TcpIpConfig/TcpIpIpConfig	
BSW Parameter		**BSW Type**
TcpIpIpReassTimeout		EcucFloatParamDef
BSW Description		
Specifies the timeout in [s] after which an incomplete datagram gets discarded. Note: this parameter is only relevant if TcpIpIpFragmentationRxEnabled is TRUE.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
TcpIp	TcpIp/TcpIpConfig	
BSW Parameter		**BSW Type**
TcpIpLocalAddr		EcucParamConfContainerDef
BSW Description		
Specifies the local IP (Internet Protocol) addresses used for IP communication.		
M2 Template	**M2 Description**	
TPS_SYST	The network endpoint defines the network addressing (e.g. IP-Address or MAC multicast address).	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::NetworkEndpointAddress		
Mapping Rule		**Mapping Type**
Create container for each NetworkEndpointAddress element that is defined in the ECU Extract.		full

BSW Module	**BSW Context**	
TcpIp	TcpIp/TcpIpConfig/TcpIpLocalAddr	
BSW Parameter		**BSW Type**
TcpIpAddrAssignment		EcucParamConfContainerDef
BSW Description		
This container is a subcontainer of TcpIpLocalAddr and specifies the assignment policy for the IP address.		

M2 Template	M2 Description	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
TcpIp	TcpIp/TcpIpConfig/TcpIpLocalAddr/TcpIpAddrAssignment	
BSW Parameter		**BSW Type**
TcpIpAssignmentMethod		EcucEnumerationParamDef
BSW Description		
Method of address assignment		
M2 Template	**M2 Description**	
TPS_SYST	Defines how the node obtains its IP address.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv4Configuration.ipv4AddressSource, SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv6Configuration.ipv6AddressSource		
Mapping Rule		**Mapping Type**
Derive parameter from the AddressSource attributes.		full

BSW Module	**BSW Context**	
TcpIp	TcpIp/TcpIpConfig/TcpIpLocalAddr/TcpIpAddrAssignment	
BSW Parameter		**BSW Type**
TcpIpAssignmentPriority		EcucIntegerParamDef
BSW Description		
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.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
TcpIp	TcpIp/TcpIpConfig/TcpIpLocalAddr/TcpIpAddrAssignment	
BSW Parameter		**BSW Type**
TcpIpAssignmentTrigger		EcucEnumerationParamDef
BSW Description		
Trigger of address assignment.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context		
TcpIp	TcpIp/TcpIpConfig/TcpIpLocalAddr		
BSW Parameter		**BSW Type**	
TcpIpAddrId		EcucIntegerParamDef	
BSW Description			
IP address table identifier assigned by TCP/IP stack.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
TcpIp	TcpIp/TcpIpConfig/TcpIpLocalAddr		
BSW Parameter		**BSW Type**	
TcpIpAddressType		EcucEnumerationParamDef	
BSW Description			
Address type.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
TcpIp	TcpIp/TcpIpConfig/TcpIpLocalAddr		
BSW Parameter		**BSW Type**	
TcpIpCtrlRef		EcucReferenceDef	
BSW Description			
Reference to a TcpIpCtrl specifying the EthIf Controller where the IP address shall be assigned and DEM errors that shall be reported in case of an error on this controller.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpLocalAddr	
BSW Parameter		**BSW Type**
TcpIpDomainType		EcucEnumerationParamDef
BSW Description		
Address family.		
M2 Template	**M2 Description**	
TPS_SYST	To build a valid network endpoint address there has to be either one MAC multicast group reference or an ipv4 configuration or an ipv6 configuration.	
M2 Parameter		

SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::NetworkEndpointAddress	
Mapping Rule	**Mapping Type**
Derive this parameter from the NetworkEndpointAddress.	full

BSW Module	**BSW Context**	
TcpIp	TcpIp/TcpIpConfig/TcpIpLocalAddr	
BSW Parameter		**BSW Type**
TcpIpStaticIpAddressConfig		EcucParamConfContainerDef
BSW Description		
This container is a subcontainer of TcpIpLocalAddr and specifies a static IP address including directly related parameters.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
TcpIp	TcpIp/TcpIpConfig/TcpIpLocalAddr/TcpIpStaticIpAddressConfig	
BSW Parameter		**BSW Type**
TcpIpDefaultRouter		EcucStringParamDef
BSW Description		
IP address of default router (gateway)		
M2 Template	**M2 Description**	
TPS_SYST	IP address of the default router (IPv6) and gateway (IPv4)	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv6Configuration.defaultRouter,SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv4Configuration.defaultGateway		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
TcpIp	TcpIp/TcpIpConfig/TcpIpLocalAddr/TcpIpStaticIpAddressConfig	
BSW Parameter		**BSW Type**
TcpIpNetmask		EcucIntegerParamDef
BSW Description		
Network mask of IPv4 address or address prefix of IPv6 address in CIDR Notation, i.e. decimal value between 0 and 32 (IPv4) or 0 and 128 (IPv6) that describes the number of significant bits defining the network number or prefix of an IP address.		
M2 Template	**M2 Description**	
TPS_SYST	Network mask. Notation 255.255.255.255	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv4Configuration.networkMask		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**
TcpIp	TcpIp/TcpIpConfig/TcpIpLocalAddr/TcpIpStaticIpAddressConfig

BSW Parameter		BSW Type	
TcpIpStaticIpAddress		EcucStringParamDef	
BSW Description			
Static IP Address. To specify any IP address for a certain EthIfCtrl, "ANY" has to be set as wildcard. See TcpIp_Bind() for more details.			
M2 Template	**M2 Description**		
TPS_SYST	IP Address (IPv4 or IPv6)		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv4Configuration.ipv4Address,SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::Ipv6Configuration.ipv6Address			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	BSW Context		
TcpIp	TcpIp/TcpIpConfig		
BSW Parameter		**BSW Type**	
TcpIpPhysAddrConfig		EcucParamConfContainerDef	
BSW Description			
Specifies the physical address configuration.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
TcpIp	TcpIp/TcpIpConfig/TcpIpPhysAddrConfig		
BSW Parameter		**BSW Type**	
TcpIpPhysAddrChgHandler		EcucParamConfContainerDef	
BSW Description			
This container is a subcontainer of TcpIpPhysAddrConfig and specifies the configuration parameters for physical address change handler.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
TcpIp	TcpIp/TcpIpConfig/TcpIpPhysAddrConfig/TcpIpPhysAddrChgHandler		
BSW Parameter		**BSW Type**	
TcpIpPhysAddrChgHandlerHeaderFileName		EcucStringParamDef	
BSW Description			
This parameter specifies the name of the header file containing the definition of the physical address change handler function.			
M2 Template	**M2 Description**		

M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpPhysAddrConfig/TcpIpPhysAddrChgHandler	
BSW Parameter		**BSW Type**
TcpIpPhysAddrChgHandlerName		EcucFunctionNameDef
BSW Description		
This parameter defines the name of the physical address change function <Up>_PhysAddrTableChg.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig	
BSW Parameter		**BSW Type**
TcpIpUdpConfig		EcucParamConfContainerDef
BSW Description		
Specifies the configuration parameters of the UDP (User Datagram Protocol) sub-module		
M2 Template	**M2 Description**	
TPS_SYST	Content Model for UDP configuration.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::UdpTp		
Mapping Rule		**Mapping Type**
This container shall be created if the UdpTp element is used in the ECU Extract.		full

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpIpUdpConfig	
BSW Parameter		**BSW Type**
TcpIpUdpTtl		EcucIntegerParamDef
BSW Description		
Default Time-to-live value of outgoing UDP packets.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig	
BSW Parameter		**BSW Type**
TcpKeepAliveProbesMax		EcucParamConfContainerDef

BSW Description	
Specifies the configuration parameters of the TCP (Transmission Control Protocol) sub-module.	
M2 Template	**M2 Description**
TPS_SYST	Content Model for TCP configuration.
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::TcpTp	
Mapping Rule	**Mapping Type**
This container shall be created if the TcpTp element is used in the ECU Extract.	full

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpKeepAliveProbesMax
BSW Parameter	**BSW Type**
TcpIpTcpCongestionAvoidanceEnabled	EcucBooleanParamDef
BSW Description	
Enables (TRUE) or disables (FALSE) support of TCP congestion avoidance algorithm according to IETF RFC 5681.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpKeepAliveProbesMax
BSW Parameter	**BSW Type**
TcpIpTcpFastRecoveryEnabled	EcucBooleanParamDef
BSW Description	
Enables (TRUE) or disables (FALSE) support of TCP Fast Recovery according to IETF RFC 5681.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpConfig/TcpKeepAliveProbesMax
BSW Parameter	**BSW Type**
TcpIpTcpFastRetransmitEnabled	EcucBooleanParamDef
BSW Description	
Enables (TRUE) or disables (FALSE) support of TCP Fast Retransmission according to IETF RFC 5681.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context		
TcpIp	TcpIp/TcpIpConfig/TcpKeepAliveProbesMax		
BSW Parameter		**BSW Type**	
TcpIpTcpFinWait2Timeout		EcucFloatParamDef	
BSW Description			
Timeout in [s] to receive a FIN from the remote node (after this node has initiated connection termination), i.e. maximum time waiting in FINWAIT-2 for a connection termination request from the remote TCP.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
TcpIp	TcpIp/TcpIpConfig/TcpKeepAliveProbesMax		
BSW Parameter		**BSW Type**	
TcpIpTcpKeepAliveEnabled		EcucBooleanParamDef	
BSW Description			
Enables (TRUE) or disables (FALSE) TCP Keep Alive Probes according to IETF RFC 1122 chapter 4.2.3.6			
M2 Template	**M2 Description**		
TPS_SYST	Indicates if Keep-Alive messages are send.		
M2 Parameter			
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::TcpTp.keepAlives			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	BSW Context		
TcpIp	TcpIp/TcpIpConfig/TcpKeepAliveProbesMax		
BSW Parameter		**BSW Type**	
TcpIpTcpKeepAliveInterval		EcucFloatParamDef	
BSW Description			
Specifies the interval in [s] between subsequent keepalive probes.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
TcpIp	TcpIp/TcpIpConfig/TcpKeepAliveProbesMax		
BSW Parameter		**BSW Type**	
TcpIpTcpKeepAliveProbesMax		EcucIntegerParamDef	
BSW Description			
Maximum number of times that a TCP segment is retransmitted.			
M2 Template	**M2 Description**		

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpKeepAliveProbesMax	
BSW Parameter		**BSW Type**
TcpIpTcpMaxRtx		EcucIntegerParamDef
BSW Description		
Maximum number of times that a TCP segment is retransmitted.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpKeepAliveProbesMax	
BSW Parameter		**BSW Type**
TcpIpTcpMsl		EcucFloatParamDef
BSW Description		
Maximum segment lifetime in [s]. (Note: TIME-WAIT = 2 x TcpIpTcpMsl - to ensure that the remote node received the acknowledgment to its connection termination request.)		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpKeepAliveProbesMax	
BSW Parameter		**BSW Type**
TcpIpTcpNagleEnabled		EcucBooleanParamDef
BSW Description		
Enables (TRUE) or disables (FALSE) support of Nagle's algorithm according to IETF RFC 896. If enabled the Nagle's algorithm is activated per default for all TCP sockets, but can be deactivated via TcpIp_ChangeParameter() API.		
M2 Template	**M2 Description**	
TPS_SYST	Indicates if Nagle's Algorithm is used.	
M2 Parameter		
SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::TcpTp.naglesAlgorithm		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context

TcpIp	TcpIp/TcpIpConfig/TcpKeepAliveProbesMax		
BSW Parameter		**BSW Type**	
TcpIpTcpReceiveWindowMax		EcucIntegerParamDef	
BSW Description			
Default value of maximum receive window in bytes.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
TcpIp	TcpIp/TcpIpConfig/TcpKeepAliveProbesMax		
BSW Parameter		**BSW Type**	
TcpIpTcpRetransmissionTimeout		EcucFloatParamDef	
BSW Description			
Timeout in [s] before an unacknowledged TCP segment is sent again.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
TcpIp	TcpIp/TcpIpConfig/TcpKeepAliveProbesMax		
BSW Parameter		**BSW Type**	
TcpIpTcpSlowStartEnabled		EcucBooleanParamDef	
BSW Description			
Enables (TRUE) or disables (FALSE) support of TCP slow start algorithm according to IETF RFC 5681.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**	
TcpIp	TcpIp/TcpIpConfig/TcpKeepAliveProbesMax	
BSW Parameter		**BSW Type**
TcpIpTcpSynMaxRtx		EcucIntegerParamDef
BSW Description		
Maximum number of times that a TCP SYN is retransmitted. Note: SYN will be retried after TcpIpTcpRetransmissionTimeout. The connection will be dropped if no matching connection request has been received after the last TCP SYN has been sent and TcpIpTcpRetransmissionTimeout has been expired.		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpKeepAliveProbesMax	
BSW Parameter		**BSW Type**
TcpIpTcpSynReceivedTimeout		EcucFloatParamDef
BSW Description		
Timeout in [s] to complete a remotely initiated TCP connection establishment, i.e. maximum time waiting in SYN-RECEIVED for a confirming connection request acknowledgment after having both received and sent a connection request.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpConfig/TcpKeepAliveProbesMax	
BSW Parameter		**BSW Type**
TcpIpTcpTtl		EcucIntegerParamDef
BSW Description		
Default Time-to-live value of outgoing TCP packets.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
TcpIp	TcpIp	
BSW Parameter		**BSW Type**
TcpIpGeneral		EcucParamConfContainerDef
BSW Description		
This container is a subcontainer of TcpIp and specifies the general configuration parameters of the TCP/IP stack.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context
TcpIp	TcpIp/TcpIpGeneral

BSW Parameter			BSW Type
TcpIpArpEnabled			EcucBooleanParamDef
BSW Description			
Enables (TRUE) or disables (FALSE) support of ARP (Address Resolution Protocol).			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
TcpIp	TcpIp/TcpIpGeneral		
BSW Parameter		**BSW Type**	
TcpIpAutoIpEnabled		EcucBooleanParamDef	
BSW Description			
Enables (TRUE) or disables (FALSE) the Auto-IP (automatic private IP addressing) sub-module.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
TcpIp	TcpIp/TcpIpGeneral		
BSW Parameter		**BSW Type**	
TcpIpBufferMemory		EcucIntegerParamDef	
BSW Description			
Memory size in bytes reserved for TCP/IP buffers.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
TcpIp	TcpIp/TcpIpGeneral		
BSW Parameter		**BSW Type**	
TcpIpDevErrorDetect		EcucBooleanParamDef	
BSW Description			
If true then TCP/IP will enable the error-reporting to the Development Error Tracer (DET).			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
TcpIp	TcpIp/TcpIpGeneral		
BSW Parameter		**BSW Type**	
TcpIpDhcpClientEnabled		EcucBooleanParamDef	
BSW Description			
Enables (TRUE) or disables (FALSE) the DHCP (Dynamic Host Configuration Protocol) Client.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
TcpIp	TcpIp/TcpIpGeneral		
BSW Parameter		**BSW Type**	
TcpIpIcmpEnabled		EcucBooleanParamDef	
BSW Description			
Enables (TRUE) or disabled (FALSE) support of ICMP (Internet Control Message Protocol).			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
TcpIp	TcpIp/TcpIpGeneral		
BSW Parameter		**BSW Type**	
TcpIpLocalAddrIpv4EntriesMax		EcucIntegerParamDef	
BSW Description			
Maximum number of LocalAddr table entries for IPv4.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
TcpIp	TcpIp/TcpIpGeneral		
BSW Parameter		**BSW Type**	
TcpIpLocalAddrIpv6EntriesMax		EcucIntegerParamDef	
BSW Description			
Maximum number of LocalAddr table entries for IPv6.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

	local

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpGeneral	
BSW Parameter		**BSW Type**
TcpIpMainFunctionPeriod		EcucFloatParamDef
BSW Description		
Period of TcpIp_MainFunction in [s].		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpGeneral	
BSW Parameter		**BSW Type**
TcpIpPathMtuDiscoveryEnabled		EcucBooleanParamDef
BSW Description		
Enables (TRUE) or disables (FALSE) the discovery of the maximum transmission unit on a path according to IETF RfC 1191.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpGeneral	
BSW Parameter		**BSW Type**
TcpIpTcpEnabled		EcucBooleanParamDef
BSW Description		
Enables (TRUE) or disabled (FALSE) support of TCP (Transmission Control Protocol).		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
TcpIp	TcpIp/TcpIpGeneral	
BSW Parameter		**BSW Type**
TcpIpTcpSocketMax		EcucIntegerParamDef
BSW Description		
Maximum number of TCP sockets		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**	
TcpIp	TcpIp/TcpIpGeneral	
BSW Parameter		**BSW Type**
TcpIpUdpEnabled		EcucBooleanParamDef
BSW Description		
Enables (TRUE) or disabled (FALSE) support of UDP (User Datagram Protocol)		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
TcpIp	TcpIp/TcpIpGeneral	
BSW Parameter		**BSW Type**
TcpIpUdpSocketMax		EcucIntegerParamDef
BSW Description		
Maximum number of UDP sockets.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
TcpIp	TcpIp/TcpIpGeneral	
BSW Parameter		**BSW Type**
TcpIpVersionInfoApi		EcucBooleanParamDef
BSW Description		
If true the TcpIp_GetVersionInfo API is available.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

//

D.6.8 DoIP

// //

BSW Module	BSW Context	
DoIP	DoIP	
BSW Parameter		**BSW Type**
DoIPConfigSet		EcucParamConfContainerDef
BSW Description		
This container contains the configuration parameters and sub containers of the AUTOSAR DoIP module. This container is a MultipleConfigurationContainer, i.e. this container and its sub-containers exist once per configuration set.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet	
BSW Parameter		**BSW Type**
DoIPChannel		EcucParamConfContainerDef
BSW Description		
Configuration of one DoIPChannel.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPChannel	
BSW Parameter		**BSW Type**
DoIPChannelSARef		EcucReferenceDef
BSW Description		
Reference to the DoIPTester.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPChannel	
BSW Parameter		**BSW Type**
DoIPChannelTARef		EcucReferenceDef
BSW Description		
Reference to the target address.		

M2 Template	M2 Description	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context		
DoIP	DoIP/DoIPConfigSet/DoIPChannel		
BSW Parameter		**BSW Type**	
DoIPPduRRxPdu		EcucParamConfContainerDef	
BSW Description			
This container contains the Rx Pdus to connect with the Rx Pdus of the PduR.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	BSW Context		
DoIP	DoIP/DoIPConfigSet/DoIPChannel/DoIPPduRRxPdu		
BSW Parameter		**BSW Type**	
DoIPPduRRxPduId		EcucIntegerParamDef	
BSW Description			
The DoIPPduRRxPduId is required by the API call DoIP_CancelReceive.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	BSW Context		
DoIP	DoIP/DoIPConfigSet/DoIPChannel/DoIPPduRRxPdu		
BSW Parameter		**BSW Type**	
DoIPPduRRxPduRef		EcucReferenceDef	
BSW Description			
Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPChannel	
BSW Parameter		**BSW Type**

DoIPPduRTxPdu	EcucParamConfContainerDef
BSW Description	
This container contains the Tx Pdus to connect with the Tx Pdus of the PduR. If the parameter is not configured the channel is for functional addressing.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**
DoIP	DoIP/DoIPConfigSet/DoIPChannel/DoIPPduRTxPdu
BSW Parameter	**BSW Type**
DoIPPduRTxPduId	EcucIntegerParamDef
BSW Description	
The DoIPPduRTxPduId is required by DoIP_Transmit and DoIP_CancelTransmit.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**
DoIP	DoIP/DoIPConfigSet/DoIPChannel/DoIPPduRTxPdu
BSW Parameter	**BSW Type**
DoIPPduRTxPduRef	EcucReferenceDef
BSW Description	
Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
DoIP	DoIP/DoIPConfigSet
BSW Parameter	**BSW Type**
DoIPConnections	EcucParamConfContainerDef
BSW Description	
Container contains all lower layer connection specific information, i.e. the single Pdu References and Handle IDs to the SoAd.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPConnections	
BSW Parameter		**BSW Type**
DoIPTargetAddress		EcucParamConfContainerDef
BSW Description		
This container describes a possible TargetAddress that is supported by DoIP.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPConnections/DoIPTargetAddress	
BSW Parameter		**BSW Type**
DoIPTargetAddressValue		EcucIntegerParamDef
BSW Description		
Valid Target Address of a DoIP target address.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPConnections	
BSW Parameter		**BSW Type**
DoIPTcpConnection		EcucParamConfContainerDef
BSW Description		
This container describes a tcp connection to the lower layer SoAd module.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPConnections/DoIPTcpConnection	
BSW Parameter		**BSW Type**
DoIPSoAdRxPdu		EcucParamConfContainerDef
BSW Description		
This container contains the Rx Pdus received by DoIP		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPConnections/DoIPTcpConnection/DoIPSoAdRxPdu	
BSW Parameter		**BSW Type**
DoIPSoAdRxPduId		EcucIntegerParamDef
BSW Description		
The DoIPSoAdRxPduId is required by the API call DoIP_SoAdTpRxIndication to receive I-PDUs from the SoAd.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPConnections/DoIPTcpConnection/DoIPSoAdRxPdu	
BSW Parameter		**BSW Type**
DoIPSoAdRxPduRef		EcucReferenceDef
BSW Description		
Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPConnections/DoIPTcpConnection	
BSW Parameter		**BSW Type**
DoIPSoAdTxPdu		EcucParamConfContainerDef
BSW Description		
This container describes the TxPdu sent via the SoAd		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPConnections/DoIPTcpConnection/DoIPSoAdTxPdu	
BSW Parameter		**BSW Type**
DoIPSoAdTxPduId		EcucIntegerParamDef
BSW Description		
The DoIPSoAdTxPduId is required by the API call DoIP_SoAdTpTxConfirmation that is called by the SoAd to confirm that the IPdu has been transmitted successfully.		

M2 Template	M2 Description	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context		
DoIP	DoIP/DoIPConfigSet/DoIPConnections/DoIPTcpConnection/DoIPSoAdTxPdu		
BSW Parameter		**BSW Type**	
DoIPSoAdTxPduRef		EcucReferenceDef	
BSW Description			
Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
DoIP	DoIP/DoIPConfigSet/DoIPConnections		
BSW Parameter		**BSW Type**	
DoIPUdpConnection		EcucParamConfContainerDef	
BSW Description			
This Container describes the udp connection to the lower layer SoAd module.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	BSW Context		
DoIP	DoIP/DoIPConfigSet/DoIPConnections/DoIPUdpConnection		
BSW Parameter		**BSW Type**	
DoIPSoAdRxPdu		EcucParamConfContainerDef	
BSW Description			
This container contains the Rx Pdus received by DoIP			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPConnections/DoIPUdpConnection/DoIPSoAdRxPdu	
BSW Parameter		**BSW Type**

DoIPSoAdRxPduId	EcucIntegerParamDef
BSW Description	
The DoIPSoAdRxPduId is required by the API call DoIP_SoAdTpRxIndication to receive I-PDUs from the SoAd.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**
DoIP	DoIP/DoIPConfigSet/DoIPConnections/DoIPUdpConnection/DoIPSoAdRxPdu
BSW Parameter	**BSW Type**
DoIPSoAdRxPduRef	EcucReferenceDef
BSW Description	
Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
DoIP	DoIP/DoIPConfigSet/DoIPConnections/DoIPUdpConnection
BSW Parameter	**BSW Type**
DoIPSoAdTxPdu	EcucParamConfContainerDef
BSW Description	
This container describes the TxPdu sent via the SoAd	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**
DoIP	DoIP/DoIPConfigSet/DoIPConnections/DoIPUdpConnection/DoIPSoAdTxPdu
BSW Parameter	**BSW Type**
DoIPSoAdTxPduId	EcucIntegerParamDef
BSW Description	
The DoIPSoAdTxPduId is required by the API call DoIP_SoAdTpTxConfirmation that is called by the SoAd to confirm that the IPdu has been transmitted successfully.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet/DoIPConnections/DoIPUdpConnection/DoIPSoAdTxPdu	
BSW Parameter		**BSW Type**
DoIPSoAdTxPduRef		EcucReferenceDef
BSW Description		
Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet	
BSW Parameter		**BSW Type**
DoIPEid		EcucIntegerParamDef
BSW Description		
Configured EID (Entity ID of) for vehicle identification/vehicle announcement. Only necessary if DoIPUseMacAddressForIdentification is set to FALSE.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet	
BSW Parameter		**BSW Type**
DoIPGid		EcucIntegerParamDef
BSW Description		
Configured GID (Group ID of) for vehicle identification/vehicle announcement.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
DoIP	DoIP/DoIPConfigSet	
BSW Parameter		**BSW Type**
DoIPLogicalAddress		EcucIntegerParamDef
BSW Description		
Describes the logical address of the DoIP entity, i.e. the LA that will route diagnostic requests to the Dcm of the DoIP entity.		
M2 Template	**M2 Description**	
M2 Parameter		

Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**	
DoIP	DoIP/DoIPConfigSet	
BSW Parameter		**BSW Type**
DoIPRoutingActivation		EcucParamConfContainerDef
BSW Description		
This container describes the routing activation possibilities by representing for each container a possible routing activation request message to the DoIP entity and the according references to the activated diagnostic messages.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
DoIP	DoIP/DoIPConfigSet/DoIPRoutingActivation	
BSW Parameter		**BSW Type**
DoIPRoutingActivationAuthenticationCallback		EcucParamConfContainerDef
BSW Description		
Container describes the Callbackfunction to call on a Routing Activation Request for Authentication. If this container is configured but the DoIPRoutingActivationAuthenticationFunc parameter is not present, the DoIP module will use an RPort of ServiceInterface <RoutingActivation>_RoutingActivation with the name "CB<RoutingActivation>RoutingActivation". <RoutingActivation> is the ShortName of the DoIPRoutingActiviation container.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
DoIP	DoIP/DoIPConfigSet/DoIPRoutingActivation/DoIPRoutingActivationAuthenticationCallback	
BSW Parameter		**BSW Type**
DoIPRoutingActivationAuthenticationFunc		EcucFunctionNameDef
BSW Description		
Direct C Callback function to trigger the authentication function for routing activation. If the DoIPRoutingActivationAuthenticationFunc parameter is present, the DoIP module will not use an RPort of ServiceInterface <RoutingActivation>_RoutingActivation but call the configured function.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context
DoIP	DoIP/DoIPConfigSet/DoIPRoutingActivation/DoIPRoutingActivationAuthenticationCallback
BSW Parameter	**BSW Type**
DoIPRoutingActivationAuthenticationReqLength	EcucIntegerParamDef
BSW Description	
Describes the amount of bytes used to handle to the authentication function on routing activation. If 0 is configured as length the parameter AuthenticationReqData will not be handled to the API.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	BSW Context
DoIP	DoIP/DoIPConfigSet/DoIPRoutingActivation/DoIPRoutingActivationAuthenticationCallback
BSW Parameter	**BSW Type**
DoIPRoutingActivationAuthenticationResLength	EcucIntegerParamDef
BSW Description	
Describes the amount of bytes used to read by the authentication function on routing activation. If 0 is configured as length the parameter AuthenticationResData will not be fetched via the API.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	BSW Context
DoIP	DoIP/DoIPConfigSet/DoIPRoutingActivation
BSW Parameter	**BSW Type**
DoIPRoutingActivationConfirmationCallback	EcucParamConfContainerDef
BSW Description	
Container describes the Callbackfunction to call on a Routing Activation Request for Confirmation. If this container is configured but the DoIPRoutingActivationConfirmationFunc parameter is not present the DoIP module will use an RPort of ServiceInterface <RoutingActivation>_RoutingActivation with the name "CB<RoutingActivation>RoutingActivation". <RoutingActivation> is the ShortName of the DoIPRoutingActiviation container.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	BSW Context
DoIP	DoIP/DoIPConfigSet/DoIPRoutingActivation/DoIPRoutingActivationConfirmationCallback
BSW Parameter	**BSW Type**

DoIPRoutingActivationConfirmationFunc		EcucFunctionNameDef
BSW Description		
Direct C Callback function to trigger the confirmation function for routing activation. If the DoIPRoutingActivationConfirmationFunc parameter is present the DoIP module will not use an RPort of ServiceInterface <RoutingActivation>_RoutingActivation but call the configured function.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
DoIP	DoIP/DoIPConfigSet/DoIPRoutingActivation/DoIPRoutingActivationConfirmationCallback	
BSW Parameter		**BSW Type**
DoIPRoutingActivationConfirmationReqLength		EcucIntegerParamDef
BSW Description		
Describes the amount of bytes used to handle to the confirmation function on routing activation. If 0 is configured as length the parameter ConfirmedReqData will not be handled to the API.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
DoIP	DoIP/DoIPConfigSet/DoIPRoutingActivation/DoIPRoutingActivationConfirmationCallback	
BSW Parameter		**BSW Type**
DoIPRoutingActivationConfirmationResLength		EcucIntegerParamDef
BSW Description		
Describes the amount of bytes used to read by the confirmation function on routing activation. If 0 is configured as length the parameter ConfirmedResData will not be fetched via the API.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
DoIP	DoIP/DoIPConfigSet/DoIPRoutingActivation	
BSW Parameter		**BSW Type**
DoIPRoutingActivationNumber		EcucIntegerParamDef
BSW Description		
Identifies the Routing activation Number which is received for a DoIP routing activation request message.		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**	
DoIP	DoIP/DoIPConfigSet/DoIPRoutingActivation	
BSW Parameter		**BSW Type**
DoIPTargetAddressRef		EcucReferenceDef
BSW Description		
Reference to all DoIPTargetAddress which are activated on this Routing activation.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
DoIP	DoIP/DoIPConfigSet	
BSW Parameter		**BSW Type**
DoIPTester		EcucParamConfContainerDef
BSW Description		
This container describes the properties of the possible connectable Tester for the DoIP entity.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
DoIP	DoIP/DoIPConfigSet/DoIPTester	
BSW Parameter		**BSW Type**
DoIPNumByteDiagAckNack		EcucIntegerParamDef
BSW Description		
Specifies the number of original Diagnostic request bytes the DoIP entity responses on a NACK of a diagnostic response message to the Tester.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
DoIP	DoIP/DoIPConfigSet/DoIPTester	
BSW Parameter		**BSW Type**

DoIPRoutingActivationRef		EcucReferenceDef	
BSW Description			
Reference to a DoIPRoutingActivation describing the possible routing activations of the DoIPTester			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	**BSW Context**		
DoIP	DoIP/DoIPConfigSet/DoIPTester		
BSW Parameter		**BSW Type**	
DoIPTesterSA		EcucIntegerParamDef	
BSW Description			
Source Address of the Tester sent via routing activation or diagnostic message.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	**BSW Context**		
DoIP	DoIP		
BSW Parameter		**BSW Type**	
DoIPGeneral		EcucParamConfContainerDef	
BSW Description			
This container specifies the general configuration parameters of the DoIP module.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	**BSW Context**		
DoIP	DoIP/DoIPGeneral		
BSW Parameter		**BSW Type**	
DoIPAliveCheckResponseTimeout		EcucFloatParamDef	
BSW Description			
Timeout in [s] for waiting for a response to an Alive Check request before the connection is considered to be disconnected. Represents parameter T_TCP_AliveCheck of ISO 13400-2:2012.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	BSW Context
DoIP	DoIP/DoIPGeneral
BSW Parameter	**BSW Type**
DoIPDevelopmentErrorDetect	EcucBooleanParamDef
BSW Description	
Pre-processor switch for enabling development error detection support.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	BSW Context
DoIP	DoIP/DoIPGeneral
BSW Parameter	**BSW Type**
DoIPDhcpOptionVinUse	EcucBooleanParamDef
BSW Description	
If DoIPDhcpOptionVinUse is set to true the DoIP module will add the VIN to the Dhcp host name if no valid Dhcp host name is already set.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	BSW Context
DoIP	DoIP/DoIPGeneral
BSW Parameter	**BSW Type**
DoIPEntityStatusMaxByteFieldUse	EcucBooleanParamDef
BSW Description	
This parameter is used to distinguish the optional support of the Max data size element of a diagnostic entity status response.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**

BSW Module	BSW Context
DoIP	DoIP/DoIPGeneral
BSW Parameter	**BSW Type**
DoIPGIDInvalidityPattern	EcucIntegerParamDef
BSW Description	
Specifies the Byte pattern that is used for response messages if no valid GID could be retrieved.	
M2 Template	**M2 Description**
M2 Parameter	

Mapping Rule	**Mapping Type**

BSW Module	**BSW Context**	
DoIP	DoIP/DoIPGeneral	
BSW Parameter		**BSW Type**
DoIPGeneralInactivityTime		EcucFloatParamDef
BSW Description		
Timeout in [s] for maximum inactivity of a TCP socket connection before the DoIP module will close the according socket connection. Represents parameter T_TCP_General_Inactivity of ISO 13400-2:2012		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
DoIP	DoIP/DoIPGeneral	
BSW Parameter		**BSW Type**
DoIPGetGid		EcucParamConfContainerDef
BSW Description		
This container describes the Callbackfunction to get the GID. If this container is not configured no Callbackfunction will be used. If this container is configured but the DoIPGetGID parameter is not present, the DoIP module will use an RPort of ServiceInterface CallbackGetGID with the name "CBGetGID".		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
DoIP	DoIP/DoIPGeneral/DoIPGetGid	
BSW Parameter		**BSW Type**
DoIPGetGID		EcucFunctionNameDef
BSW Description		
Direct C Callback function to get the GID of the DoIP entity. If the DoIPGetGID parameter is present, the DoIP module will not use an RPort of ServiceInterface CallbackGetGID but call the configured function.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
DoIP	DoIP/DoIPGeneral	
BSW Parameter		**BSW Type**
DoIPHeaderFileInclusion		EcucStringParamDef
BSW Description		
Name of the header file(s) to be included by the DoIP module containing the used C-callback declarations.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
DoIP	DoIP/DoIPGeneral	
BSW Parameter		**BSW Type**
DoIPHostNameSizeMax		EcucStringParamDef
BSW Description		
Maximum Size of the DHCP HostName in ASCII. This parameter is necessary to reserve the correct amount of bytes for working with the DHCP HostName option. Minimum range is 5 because Dhcp Host Name should be at least "DoIP-" on any configuration.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
DoIP	DoIP/DoIPGeneral	
BSW Parameter		**BSW Type**
DoIPInitialInactivityTime		EcucFloatParamDef
BSW Description		
Timeout in [s] used for initial inactivity of a connected TCP socket connection directly after socket connection. Represents parameter T_TCP_Initial_Inactivity of ISO 13400-2:2012		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
DoIP	DoIP/DoIPGeneral	
BSW Parameter		**BSW Type**
DoIPInitialVehicleAnnouncementTime		EcucFloatParamDef
BSW Description		
Time to wait in [s] for sending first vehicle anouncement message after IP address assignment. Represents parameter A_DoIP_Announce_Wait of ISO 13400-2:2012		

M2 Template	M2 Description	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
DoIP	DoIP/DoIPGeneral	
BSW Parameter		**BSW Type**
DoIPMainFunctionPeriod		EcucFloatParamDef
BSW Description		
Determines the frequency at which the DoIP_MainFunction() is called in [s].		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
DoIP	DoIP/DoIPGeneral	
BSW Parameter		**BSW Type**
DoIPMaxRequestBytes		EcucIntegerParamDef
BSW Description		
Specifies the maximum allowed bytes of a DoIP message request without the DoIP header.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
DoIP	DoIP/DoIPGeneral	
BSW Parameter		**BSW Type**
DoIPMaxTesterConnections		EcucIntegerParamDef
BSW Description		
Maximum ammount of tester connections that shall be maintained at one time before alive check is performed.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context
DoIP	DoIP/DoIPGeneral

BSW Parameter			BSW Type	
DoIPNodeType			EcucEnumerationParamDef	
BSW Description				
Describes the Type of the DoIP node.				
M2 Template	**M2 Description**			
M2 Parameter				
Mapping Rule				**Mapping Type**

BSW Module	BSW Context			
DoIP	DoIP/DoIPGeneral			
BSW Parameter			**BSW Type**	
DoIPPowerModeCallback			EcucParamConfContainerDef	
BSW Description				
This container describes the Callbackfunction to get the Power Mode. This container shall always be present. If the DoIPPowerMode parameter is not present, the DoIP module will use an RPort of ServiceInterface CallbackGetPowerMode with the name "CBGetPowerMode".				
M2 Template	**M2 Description**			
M2 Parameter				
Mapping Rule				**Mapping Type**

BSW Module	BSW Context			
DoIP	DoIP/DoIPGeneral/DoIPPowerModeCallback			
BSW Parameter			**BSW Type**	
DoIPPowerMode			EcucFunctionNameDef	
BSW Description				
Direct C Callback function to get the Power Mode of the DoIP entity. If the DoIPPowerMode parameter is present, the DoIP module will not use an RPort of ServiceInterface CallbackGetPowerMode but will call the configured function.				
M2 Template	**M2 Description**			
M2 Parameter				
Mapping Rule				**Mapping Type**

BSW Module	BSW Context		
DoIP	DoIP/DoIPGeneral		
BSW Parameter		**BSW Type**	
DoIPTriggerGidSynchronization		EcucParamConfContainerDef	
BSW Description			

This container describes the Callbackfunction to trigger the GID synchronisation. If this container is not configured no Callbackfunction will be used. If this container is configured but the DoIPTriggerGIDSynchronzation parameter is not present, the DoIP module will use an RPort of ServiceInterface CallbackTriggerGIDSynchronization with the name "CBTriggerGIDSynchronization".		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
DoIP	DoIP/DoIPGeneral/DoIPTriggerGidSynchronization	
BSW Parameter		**BSW Type**
DoIPTriggerGIDSynchronzation		EcucFunctionNameDef
BSW Description		
Direct C Callback function to trigger the synchronization of the GID. If the DoIPTriggerGIDSynchronzation parameter is present, the DoIP module will not use an RPort of ServiceInterface CallbackTriggerGIDSynchronization but call the configured function.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
DoIP	DoIP/DoIPGeneral	
BSW Parameter		**BSW Type**
DoIPUseEIDasGID		EcucBooleanParamDef
BSW Description		
Specifies if the DoIP entity shall use its EID if it is the Master for vehicle identification gid on the vehicle identification/vehicle announcement.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
DoIP	DoIP/DoIPGeneral	
BSW Parameter		**BSW Type**
DoIPUseMacAddressForIdentification		EcucBooleanParamDef
BSW Description		

Provided the information if a configured EID at vehicle identification response/vehicle announment is used or the MAC address. TRUE: Use MAC Address instead of EID for Vehicle identification/announcement. FALSE: Use configured EID for vehicle identification/announcement. Dependencies: DoIPEID		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
DoIP	DoIP/DoIPGeneral	
BSW Parameter		**BSW Type**
DoIPUseVehicleIdentificationSyncStatus		EcucBooleanParamDef
BSW Description		
Defines if the optional VIN/GID synchronization status is used additionally in the vehicle identification/announcement.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
DoIP	DoIP/DoIPGeneral	
BSW Parameter		**BSW Type**
DoIPVehicleAnnouncementInterval		EcucFloatParamDef
BSW Description		
Time to wait in [s] for sending subsequent vehicle anouncement messages. Represents parameter A_DoIP_Announce_Interval of ISO 13400-2:2012		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	**BSW Context**	
DoIP	DoIP/DoIPGeneral	
BSW Parameter		**BSW Type**
DoIPVehicleAnnouncementRepetition		EcucIntegerParamDef
BSW Description		
Amount of repetitions of the vehicle announcement message on IP address assignment. Represents parameter A_DoIP_Annunce_Num of ISO 13400-2:2012		
M2 Template	**M2 Description**	
M2 Parameter		

Mapping Rule	Mapping Type

BSW Module	BSW Context	
DoIP	DoIP/DoIPGeneral	
BSW Parameter		**BSW Type**
DoIPVersionInfoApi		EcucBooleanParamDef
BSW Description		
Activates the DoIP_GetVersionInfo() API. TRUE: Enables the DoIP_GetVersionInfo() API. FALSE: DoIP_GetVersionInfo() API is not included.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
DoIP	DoIP/DoIPGeneral	
BSW Parameter		**BSW Type**
DoIPVinGidMaster		EcucBooleanParamDef
BSW Description		
Specifies if the DoIP entity is the Vehicle identification Master for the GID (Group ID).		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

BSW Module	BSW Context	
DoIP	DoIP/DoIPGeneral	
BSW Parameter		**BSW Type**
DoIPVinInvalidityPattern		EcucIntegerParamDef
BSW Description		
Specifies the Byte pattern that is used for response messages if no valid VIN could be retrieved.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

D.6.9 UdpNm

BSW Module	BSW Context
UdpNm	UdpNm

BSW Parameter			BSW Type
UdpNmGlobalConfig			EcucParamConfContainerDef
BSW Description			
This container contains all global configuration parameters of UDP NM configured from the NM Module perspective.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
UdpNm	UdpNm/UdpNmGlobalConfig		
BSW Parameter			**BSW Type**
UdpNmBusSynchronizationEnabled			EcucBooleanParamDef
BSW Description			
Pre-processor switch for enabling bus synchronization support. This feature is required for gateway nodes only. It must not be defined if UDPNM_PASSIVE_MODE_ENABLED is defined. This parameter shall be derived from NM_BUS_SYNCHRONIZATION_ENABLED.			
M2 Template	**M2 Description**		
TPS_SYST	Enables bus synchronization support.		
M2 Parameter			
SystemTemplate::NetworkManagement::NmEcu.nmBusSynchronizationEnabled			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	BSW Context		
UdpNm	UdpNm/UdpNmGlobalConfig		
BSW Parameter			**BSW Type**
UdpNmChannelConfig			EcucParamConfContainerDef
BSW Description			
This container contains the channel-specific configuration parameters of the UdpNm.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

BSW Module	BSW Context		
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig		
BSW Parameter			**BSW Type**
UdpNmActiveWakeupBitEnabled			EcucBooleanParamDef
BSW Description			
Enables/Disables the handling of the Active Wakeup Bit in the UdpNm module.			
M2 Template	**M2 Description**		

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter		**BSW Type**
UdpNmComMNetworkHandleRef		EcucSymbolicNameReferenceDef
BSW Description		
This reference points to the unique channel defined by the ComMChannel and provides access to the unique channel index value in ComMChannelId.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter		**BSW Type**
UdpNmImmediateNmTransmissions		EcucIntegerParamDef
BSW Description		
Defines the number of immediate NM PDUs which shall be transmitted. If the value is zero no immediate NM PDUs are transmitted.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter		**BSW Type**
UdpNmMainFunctionPeriod		EcucFloatParamDef
BSW Description		
Call cycle of UdpNm_MainFunction_x for the respective instance in [s].		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter		**BSW Type**

UdpNmMsgCycleOffset	EcucFloatParamDef
BSW Description	
Time offset in the periodic transmission node. It determines the start delay of the transmission. < UDPNM_MSG_CYCLE_TIME This parameter is only valid if UDPNM_PASSIVE_MODE_ENABLED is disabled.	
M2 Template	**M2 Description**
TPS_SYST	Node specific time offset in the periodic transmission node. It determines the start delay of the transmission. Specified in seconds.
M2 Parameter	
SystemTemplate::NetworkManagement::UdpNmNode.nmMsgCycleOffset	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig
BSW Parameter	**BSW Type**
UdpNmMsgCycleTime	EcucFloatParamDef
BSW Description	
Period of a NM-message. It determines the periodic rate and is the basis for transmit scheduling. NM_TIMEOUT_TIME = n * UDPNM_MSG_CYCLE_TIME This parameter is only valid if UDPNM_PASSIVE_MODE_ENABLED is disabled.	
M2 Template	**M2 Description**
TPS_SYST	Period of a UdpNm message 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.
M2 Parameter	
SystemTemplate::NetworkManagement::UdpNmCluster.nmMsgCycleTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig
BSW Parameter	**BSW Type**
UdpNmMsgTimeoutTime	EcucFloatParamDef
BSW Description	
Transmission Timout of NM-message. If there is no transmission confirmation by the UDP Interface within this timeout, the UDPNM module shall gibe an error notification. This parameter is only valid if UDPNM_PASSIVE_MODE_ENABLED is disabled. UDPNM_MSG_TIMEOUT_TIME should be a multiple of UDPNM_MSG_CYCLE_TIME.	
M2 Template	**M2 Description**
TPS_SYST	Timeout of a NM message in seconds. It determines how long the NM shall wait with notification of transmission failure while communication errors occur on the bus.
M2 Parameter	
SystemTemplate::NetworkManagement::UdpNmCluster.nmMessageTimeoutTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig
BSW Parameter	**BSW Type**
UdpNmNodeId	EcucIntegerParamDef
BSW Description	
Node identifier of local node. This parameter is only valid if UDPNM_PASSIVE_MODE_ENABLED is set to OFF and UDPNM_NODE_DETECTION_ENABLED is set to ON.	
M2 Template	**M2 Description**
TPS_SYST	Node identifier of local NmNode. Must be unique in the NmCluster.
M2 Parameter	
SystemTemplate::NetworkManagement::NmNode.nmNodeId	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig
BSW Parameter	**BSW Type**
UdpNmPduCbvPosition	EcucEnumerationParamDef
BSW Description	

Defines the position of the control bit vector within the NM PACKET.
 	
The value of the parameter represents the location of the control bit vector in the NM PACKET (UDPNM_PDU_BYTE_0 means byte 0, UDPNM_PDU_BYTE_1 means byte 1, UDPNM_PDU_OFF means the control bit vector is not part of the NM PACKET)
 	
See also UDPNM_PDU_NID_POSITION
 	
if (UDPNM_PDU_CBV_POSITION != UDPNM_PDU_OFF && UDPNM_PDU_NID_POSITION != UDPNM_PDU_OFF) then UDPNM_PDU_CBV_POSITION != UDPNM_PDU_NID_POSITION
 	
if (UDPNM_PDU_CBV_POSITION != UDPNM_PDU_OFF && UDPNM_PDU_NID_POSITION == UDPNM_PDU_OFF) then UDPNM_PDU_CBV_POSITION = UDPNM_PDU_BYTE0	

M2 Template	**M2 Description**
TPS_SYST	Defines the position of the control bit vector within the NM PDU (Byte positon).
M2 Parameter	
SystemTemplate::NetworkManagement::UdpNmCluster.nmCbvPosition	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	BSW Context
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig
BSW Parameter	**BSW Type**
UdpNmPduLength	EcucIntegerParamDef
BSW Description	
Defines the length of the NM PACKET in bytes. Valid values are within the range 0 <= UDPNM_PDU_LENGTH <= 8.	
M2 Template	**M2 Description**
TPS_SYST	Defines the length of the NM PDU (in bytes).
M2 Parameter	

SystemTemplate::Fibex::FibexCore::CoreCommunication::Pdu.length	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter		**BSW Type**
UdpNmPduNidPosition		EcucEnumerationParamDef
BSW Description		

Defines the position of the source node identifier within the NM PACKET.
 	
ImplementationType: UdpNm_PduPositionType
 	
The value of the parameter represents the location of the source node identifier in the NM PACKET (UDPNM_PDU_BYTE_0 means byte 0, UDPNM_PDU_BYTE_1 means byte 1, UDPNM_PDU_OFF means source node identifier is not part of the NM PACKET)
 	
See also UDPNM_PDU_CBV_POSITION
 	
if (UDPNM_PDU_NID_POSITION != UDPNM_PDU_OFF && UDPNM_PDU_CBV_POSITION != UDPNM_PDU_OFF) then UDPNM_PDU_NID_POSITION != UDPNM_PDU_CBV_POSITION
 	
if (UDPNM_PDU_NID_POSITION != UDPNM_PDU_OFF && UDPNM_PDU_CBV_POSITION == UDPNM_PDU_OFF) then UDPNM_PDU_IND_POSITION = UDPNM_PDU_BYTE0		

M2 Template	**M2 Description**	
TPS_SYST	Defines the byte position of the source node identifier within the NM PDU.	
M2 Parameter		
SystemTemplate::NetworkManagement::UdpNmCluster.nmNidPosition		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter		**BSW Type**
UdpNmPnEnabled		EcucBooleanParamDef
BSW Description		
Enables or disables support of partial networking. false: Partial networking Range not supported true: Partial networking supported		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter		**BSW Type**
UdpNmPnEraCalcEnabled		EcucBooleanParamDef
BSW Description		

Specifies if UdpNm calculates the PN request information for external requests. (ERA) false: PN request are not calculated true: PN request are calculated.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig
BSW Parameter	**BSW Type**
UdpNmPnEraRxNSduRef	EcucReferenceDef
BSW Description	
Reference to a Pdu in the COM-Stack. The SduRef is required for every UdpNm Channel, because ERA is reported per channel.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig
BSW Parameter	**BSW Type**
UdpNmPnHandleMultipleNetworkRequests	EcucBooleanParamDef
BSW Description	
false: UdpNm_NetworkRequest is ignored in NO. true: UdpNm_NetworkRequest triggers a change from NO to RM.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig
BSW Parameter	**BSW Type**
UdpNmRemoteSleepIndTime	EcucFloatParamDef
BSW Description	
Timeout for Remote Sleep Indication. It defines the time in [s] how long it shall take to recognize that all other nodes are ready to sleep. Typically it should be equal to: n * UDPNM_MSG_CYCLE_TIME, where n denotes the number of NM packets that are normally sent before Remote Sleep Indication is detected. The value of n decremented by one determines the amount of lost NM packets that can be tolerated by the Remote Sleep Indication procedure.	
M2 Template	**M2 Description**

TPS_SYST	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.	
M2 Parameter		
SystemTemplate::NetworkManagement::UdpNmCluster.nmRemoteSleepIndicationTime		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter		**BSW Type**
UdpNmRepeatMessageTime		EcucFloatParamDef
BSW Description		
Timeout for Repeat Message State. It defines the time in [s] how long the NM shall stay in the Repeat Message State. Typically it should be equal to: n * UDPNM_MSG_CYCLE_TIME, where n denotes the number of NM packets that are normally sent in the Repeat Message State. The value of n decremented by one determines the amount of lost NM packets that can be tolerated by the node detection procedure. The value 0 denotes that no Repeat Message State is configured. It means that Repeat Message State is transient what implicates that it is left immediately after entrance and in result no start-up stability is guaranteed and no node detection procedure is possible.		
M2 Template	**M2 Description**	
TPS_SYST	Timeout for Repeat Message State in seconds. Defines the time how long the NM shall stay in the Repeat Message State.	
M2 Parameter		
SystemTemplate::NetworkManagement::UdpNmCluster.nmRepeatMessageTime		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter		**BSW Type**
UdpNmRxPdu		EcucParamConfContainerDef
BSW Description		
This container describes the UdpNm RX PDU's.		
M2 Template	**M2 Description**	
TPS_SYST	receive NM Pdu.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmNode.rxNmPdu		
Mapping Rule		**Mapping Type**
Create container for each NmPdu that is received on the regarded Nm cluster		full

BSW Module	**BSW Context**	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig/UdpNmRxPdu	
BSW Parameter		**BSW Type**
UdpNmRxPduId		EcucIntegerParamDef
BSW Description		
ID of the RxPdu that will be used by a RxIndication of the lower layer.		
M2 Template	**M2 Description**	

M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig/UdpNmRxPdu	
BSW Parameter		**BSW Type**
UdpNmRxPduRef		EcucReferenceDef
BSW Description		
The reference to a PDU in the global PDU structure described in the AUTOSAR ECU Configuration Specification. This reference will be used by the UdpNm module to derive the PDU Id.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter		**BSW Type**
UdpNmTimeoutTime		EcucFloatParamDef
BSW Description		
Network Timeout for NM packets. It denotes the time in [s] how long the NM shall stay in the Network Mode before transition into Prepare Bus-Sleep Mode shall take place. It shall be equal for all nodes in the cluster. It shall be greater than UDPNM_MSG_CYCLE_TIME. Typically, it should be equal to: x * UDPNM_MSG_CYCLE_TIME, where n denotes the number of NM PACKET cycle times in the Ready Sleep State before transition into the Bus-Sleep Mode is initiated. The value of n decremented by one determines the amount of lost NM packets that can be tolerated by the coordination algorithm.		
M2 Template	**M2 Description**	
TPS_SYST	Network Timeout for UdpNm PDUs 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.	
M2 Parameter		
SystemTemplate::NetworkManagement::UdpNmCluster.nmNetworkTimeout		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter		**BSW Type**
UdpNmTxPdu		EcucParamConfContainerDef
BSW Description		
This container describes the UdpNm TX PDU's.		

M2 Template	M2 Description	
TPS_SYST	transmit NM Pdu	
M2 Parameter		
SystemTemplate::NetworkManagement::NmNode.txNmPdu		
Mapping Rule		**Mapping Type**
Create container for each NmPdu that is transmitted on the regarded Nmcluster		full

BSW Module	BSW Context		
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig/UdpNmTxPdu		
BSW Parameter		**BSW Type**	
UdpNmTxConfirmationPduId		EcucIntegerParamDef	
BSW Description			
Id of the TxPdu that will be used by a TxConfirmation from the lower layer.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig/UdpNmTxPdu		
BSW Parameter		**BSW Type**	
UdpNmTxPduRef		EcucReferenceDef	
BSW Description			
The reference to a PDU in the global PDU structure described in the AUTOSAR ECU Configuration Specification. This reference will be used by the UdpNm module to derive the PDU Id.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter		**BSW Type**
UdpNmUserDataLength		EcucIntegerParamDef
BSW Description		
Please note that this parameter is deprecated and will be removed in future. Old description: Defines the length of the user data contained in the NM PACKET. The difference between UDPNM_PDU_LENGTH and applied standardized bytes (source node identifier and control bit vector) within the NM PACKET. Valid values are 0x00..0x08.		
M2 Template	**M2 Description**	
TPS_SYST	Defines the length of the user data contained in the NM Pdu.	
M2 Parameter		

SystemTemplate::NetworkManagement::UdpNmCluster.nmUserDataLength	
Mapping Rule	**Mapping Type**
1:1 mapping (Please note that this upstream mapping is deprecated)	full

BSW Module	**BSW Context**	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter		**BSW Type**
UdpNmUserDataTxPdu		EcucParamConfContainerDef
BSW Description		
This optional container is used to configure the UserNm PDU. This container is only available if UdpNmComUserDataSupport is enabled.		
M2 Template	**M2 Description**	
TPS_SYST	This optional aggregation is used to describe NmUserData 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.	
M2 Parameter		
SystemTemplate::Fibex::FibexCore::CoreCommunication::NmPdu.iSignalToIPduMapping		
Mapping Rule		**Mapping Type**
Create container for each NmPdu that aggregates the ISignalToIPduMapping element. The configuration for these Pdus (e.g. Transfer Properties) shall be derived from this information.		full

BSW Module	**BSW Context**	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig/UdpNmUserDataTxPdu	
BSW Parameter		**BSW Type**
UdpNmTxUserDataPduId		EcucIntegerParamDef
BSW Description		
This parameter defines the Handle ID of the NM User Data I-PDU.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig/UdpNmUserDataTxPdu	
BSW Parameter		**BSW Type**
UdpNmTxUserDataPduRef		EcucReferenceDef
BSW Description		
Reference to the NM User Data I-PDU in the global PDU collection.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**

UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmChannelConfig	
BSW Parameter		**BSW Type**
UdpNmWaitBusSleepTime		EcucFloatParamDef
BSW Description		
Timeout for bus calm down phase. It denotes the time in [s] how long the NM shall stay in the Prepare Bus-Sleep Mode before transition into Bus-Sleep Mode shall take place. It shall be equal for all nodes in the cluster. It shall be long enough to empty all Tx-buffer empty.		
M2 Template	**M2 Description**	
TPS_SYST	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.	
M2 Parameter		
SystemTemplate::NetworkManagement::UdpNmCluster.nmWaitBusSleepTime		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
UdpNm	UdpNm/UdpNmGlobalConfig	
BSW Parameter		**BSW Type**
UdpNmComControlEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling the Communication Control support. This parameter shall be derived from NM_COM_CONTROL_ENABLED.		
M2 Template	**M2 Description**	
TPS_SYST	Enables the Communication Control support.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmComControlEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	**BSW Context**	
UdpNm	UdpNm/UdpNmGlobalConfig	
BSW Parameter		**BSW Type**
UdpNmComUserDataSupport		EcucBooleanParamDef
BSW Description		
Enable/disable the user data support.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		full

BSW Module	**BSW Context**	
UdpNm	UdpNm/UdpNmGlobalConfig	
BSW Parameter		**BSW Type**
UdpNmCoordinatorEnabled		EcucBooleanParamDef

BSW Description		
Enable/disable the NM Coordination algorithm to being able to initiate the synchronization algorithm. TRUE: Option is enabled FALSE: The parameter shall be FALSE by default and shall only be allowed to be TRUE if the parameter UDPNM_REMOTE_SLEEP_IND_ENABLED is TRUE.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context		
UdpNm	UdpNm/UdpNmGlobalConfig		
BSW Parameter		**BSW Type**	
UdpNmCoordinatorId		EcucIntegerParamDef	
BSW Description			
Set the NM coordination ID for this gateway. 0x00: passive coordinator only 0x01 - 0x03: coordinator priority Only valid, if UDPNM_COORDINATOR_ENABLED is TRUE.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
UdpNm	UdpNm/UdpNmGlobalConfig		
BSW Parameter		**BSW Type**	
UdpNmCoordinatorSyncSupport		EcucBooleanParamDef	
BSW Description			
Enables/disables the coordinator synchronization support.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig	
BSW Parameter		**BSW Type**
UdpNmDemEventParameterRefs		EcucParamConfContainerDef
BSW Description		

Container for the references to DemEventParameter elements which shall be invoked using the API Dem_ReportErrorStatus API in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId value. The standardized errors are provided in the container and can be extended by vendor specific error references.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmDemEventParameterRefs	
BSW Parameter		**BSW Type**
UDPNM_E_NETWORK_TIMEOUT		EcucSymbolicNameReferenceDef
BSW Description		
Reference to the DemEventParameter which shall be issued when the error "NM-Timeout Timer has abnormally expired outside of the Ready Sleep State" has occured.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmDemEventParameterRefs	
BSW Parameter		**BSW Type**
UDPNM_E_TCPIP_TRANSMIT_ERROR		EcucSymbolicNameReferenceDef
BSW Description		
Reference to the DemEventParameter which shall be issued when the error "A call to the TCP/IP stack has failedA call to the TCP/IP stack has failed" has occured.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	**BSW Context**	
UdpNm	UdpNm/UdpNmGlobalConfig	
BSW Parameter		**BSW Type**
UdpNmDevErrorDetect		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling development error detection support.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**

	local

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig	
BSW Parameter		**BSW Type**
UdpNmImmediateRestartEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling the asynchronous transmission of a NM PACKET upon bus-communication request in Prepare-Bus-Sleep mode. Must not be defined if UDPNM_PASSIVE_MODE_ENABLED is defined.		
M2 Template	**M2 Description**	
TPS_SYST	Enables the asynchronous transmission of a CanNm PDU upon bus-communication request in Prepare-Bus-Sleep mode.	
M2 Parameter		
SystemTemplate::NetworkManagement::UdpNmClusterCoupling.nmImmediateRestartEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig	
BSW Parameter		**BSW Type**
UdpNmNodeDetectionEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling the node detection support. This parameter shall be derived from NM_NODE_DETECTION_ENABLED. This parameter shall only be enabled if UDPNM_NODE_ID_ENABLED is defined. If(UdpNmPduCbvPosition != UDPNM_PDU_OFF) then Equal(NmNodeDetectionEnabled) else Equal(False).		
M2 Template	**M2 Description**	
TPS_SYST	Enables the Request Repeat Message Request support. Only valid if nmNodeIdEnabled is set to true.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmNodeDetectionEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig	
BSW Parameter		**BSW Type**
UdpNmNodeIdEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling the source node identifier. This parameter shall be derived from NM_NODE_ID_ENABLED.		
M2 Template	**M2 Description**	
TPS_SYST	Enables the source node identifier.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmNodeIdEnabled		

Mapping Rule	Mapping Type
1:1 mapping	full

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig	
BSW Parameter		**BSW Type**
UdpNmNumberOfChannels		EcucIntegerParamDef
BSW Description		
Number of NM channels allowed within one ECU.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig	
BSW Parameter		**BSW Type**
UdpNmPassiveModeEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling support of the Passive Mode. This parameter shall be derived from NM_PASSIVE_MODE_ENABLED.		
M2 Template	**M2 Description**	
TPS_SYST	Enables support of the Passive Mode.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmPassiveModeEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig	
BSW Parameter		**BSW Type**
UdpNmPduRxIndicationEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling the PDU Rx Indication. This parameter shall be derived from NM_PDU_RX_INDICATION_ENABLED.		
M2 Template	**M2 Description**	
TPS_SYST	Switch for enabling the PDU Rx Indication.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmPduRxIndicationEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig	
BSW Parameter		**BSW Type**

UdpNmPnEiraCalcEnabled		EcucBooleanParamDef	
BSW Description			
Specifies if UdpNm calculates the PN request information for internal and external requests. (EIRA) true: PN request are calculated false: PN request are not calculated			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
UdpNm	UdpNm/UdpNmGlobalConfig		
BSW Parameter		**BSW Type**	
UdpNmPnEiraRxNSduRef		EcucReferenceDef	
BSW Description			
Reference to a Pdu in the COM-Stack. Only one SduRef is required for UdpNm because the EIRA is the aggregation over all Ethernet Channels.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
UdpNm	UdpNm/UdpNmGlobalConfig		
BSW Parameter		**BSW Type**	
UdpNmPnInfo		EcucParamConfContainerDef	
BSW Description			
PN information configuration			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	**BSW Context**		
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmPnInfo		
BSW Parameter		**BSW Type**	
UdpNmPnFilterMaskByte		EcucParamConfContainerDef	
BSW Description			
PN information configuration			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**

	local

BSW Module	BSW Context		
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmPnInfo/UdpNmPnFilterMaskByte		
BSW Parameter		**BSW Type**	
UdpNmPnFilterMaskByteIndex		EcucIntegerParamDef	
BSW Description			
Index of the filter mask byte. Specifies the position within the filter mask byte array.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmPnInfo/UdpNmPnFilterMaskByte		
BSW Parameter		**BSW Type**	
UdpNmPnFilterMaskByteValue		EcucIntegerParamDef	
BSW Description			
Parameter to configure the filter mask byte.			
M2 Template	**M2 Description**		
M2 Parameter			
Mapping Rule			**Mapping Type**
			local

BSW Module	BSW Context		
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmPnInfo		
BSW Parameter		**BSW Type**	
UdpNmPnInfoLength		EcucIntegerParamDef	
BSW Description			
Specifies the length of the PN request information in the NM message.			
M2 Template	**M2 Description**		
TPS_SYST	Length of the partial networking request release information vector.		
M2 Parameter			
SystemTemplate::System.pncVectorLength			
Mapping Rule			**Mapping Type**
1:1 mapping			full

BSW Module	BSW Context		
UdpNm	UdpNm/UdpNmGlobalConfig/UdpNmPnInfo		
BSW Parameter		**BSW Type**	
UdpNmPnInfoOffset		EcucIntegerParamDef	
BSW Description			
Specifies the offset of the PN request information in the NM message.			
M2 Template	**M2 Description**		

TPS_SYST	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.
M2 Parameter	
SystemTemplate::System.pncVectorOffset	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
UdpNm	UdpNm/UdpNmGlobalConfig
BSW Parameter	**BSW Type**
UdpNmPnResetTime	EcucFloatParamDef
BSW Description	
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. The value shall be the same for every channel. Thus it is a global config parameter.	
M2 Template	**M2 Description**
M2 Parameter	
Mapping Rule	**Mapping Type**
	local

BSW Module	**BSW Context**
UdpNm	UdpNm/UdpNmGlobalConfig
BSW Parameter	**BSW Type**
UdpNmRemoteSleepIndEnabled	EcucBooleanParamDef
BSW Description	
Pre-processor switch for enabling remote sleep indication support. This feature is required for gateway nodes only. It must not be defined if UDPNM_PASSIVE_MODE_ENABLED is defined. This parameter shall be derived from NM_REMOTE_SLEEP_IND_ENABLED.	
M2 Template	**M2 Description**
TPS_SYST	Switch for enabling remote sleep indication support.
M2 Parameter	
SystemTemplate::NetworkManagement::NmEcu.nmRemoteSleepIndEnabled	
Mapping Rule	**Mapping Type**
1:1 mapping	full

BSW Module	**BSW Context**
UdpNm	UdpNm/UdpNmGlobalConfig
BSW Parameter	**BSW Type**
UdpNmRepeatMsgIndEnabled	EcucBooleanParamDef
BSW Description	
Enable/disable the notification that a RepeatMessageRequest bit has been received. This parameter shall be derived from NM_REPEAT_MSG_IND_ENABLED.	
M2 Template	**M2 Description**
TPS_SYST	Switch for enabling the Repeat Message Bit Indication.
M2 Parameter	
SystemTemplate::NetworkManagement::NmEcu.nmRepeatMsgIndEnabled	

Mapping Rule	Mapping Type
1:1 mapping	full

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig	
BSW Parameter		**BSW Type**
UdpNmStateChangeIndEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling the UDP NM state change notification. This parameter shall be derived from NM_STATE_CHANGE_IND_ENABLED.		
M2 Template	**M2 Description**	
TPS_SYST	Enables the CAN Network Management state change notification.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmStateChangeIndEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig	
BSW Parameter		**BSW Type**
UdpNmUserDataEnabled		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling user data support. This parameter shall be derived from NM_USER_DATA_ENABLED.		
M2 Template	**M2 Description**	
TPS_SYST	Switch for enabling user data support.	
M2 Parameter		
SystemTemplate::NetworkManagement::NmEcu.nmUserDataEnabled		
Mapping Rule		**Mapping Type**
1:1 mapping		full

BSW Module	BSW Context	
UdpNm	UdpNm/UdpNmGlobalConfig	
BSW Parameter		**BSW Type**
UdpNmVersionInfoApi		EcucBooleanParamDef
BSW Description		
Pre-processor switch for enabling version info API support.		
M2 Template	**M2 Description**	
M2 Parameter		
Mapping Rule		**Mapping Type**
		local

E Renamed Meta-Model Elements

E.1 Introduction

In the course of preparing AUTOSAR Release 4.0 some of the existing meta-model elements have been renamed for a better clarity and consistency with respect to other meta-mode elements. This chapter provides an overview of the changed meta-model elements in order to allow readers with a background in former specifications to understand changes made by mere renaming.

E.2 Renamed Meta-Model Elements

Old Name	*New Name*
AbsolutelyScheduledTiming	FlexRayAbsolutelyScheduledTiming
IPduGroup	ISignalIPduGroup
SignalPort	ISignalPort
SignalComposition	RootSwCompositionPrototype
IPduTriggering	PduTriggering

Table E.1: Renamed meta-model elements

F Constraint History

F.1 Constraint History of this Document according to AUTOSAR R4.0.1

F.1.1 Changed Constraints in R4.0.1

N/A

F.1.2 Added Constraints in R4.0.1

Number	Heading
[constr_3000]	valid SenderRecCompositeTypeMappings
[constr_3001]	valid ClientServerToSignalGroupMappings
[constr_3002]	valid SwcToImplMapping
[constr_3003]	Number of CAN channels
[constr_3004]	Clustering and separation must be exclusive
[constr_3005]	valid EcuResourceEstimation
[constr_3006]	valid EcuMapping
[constr_3007]	SelectorFieldCodes for dynamic part alternatives
[constr_3008]	EcuInstance subelements
[constr_3009]	Overlapping of ISignals is prohibited
[constr_3010]	ISignalIPdu shall not be exceeded
[constr_3011]	Overlapping of updateIndicationBits for ISignals is prohibited
[constr_3012]	Overlapping of Pdus is prohibited
[constr_3013]	Frame length shall not be exceeded
[constr_3014]	Overlapping of updateIndicationBits for Pdus is prohibited
[constr_3015]	Number of LIN channels
[constr_3016]	Number of Ethernet channels
[constr_3017]	Length of multiplexed Pdu shall not be exceeded
[constr_3018]	Number of FlexRay channels

Table F.1: Added Constraints in R4.0.1

F.1.3 Deleted Constraints in R4.0.1

N/A

F.2 Constraint History of this Document according to AUTOSAR R4.0.2

F.2.1 Changed Constraints in R4.0.2

N/A

F.2.2 Added Constraints in R4.0.2

Number	Heading
[constr_3019]	In the flat ECU extract each required interface must be satisfied by connected provided interfaces

Table F.2: Added Constraints in R4.0.2

F.2.3 Deleted Constraints in R4.0.2

N/A

F.3 Constraint and Specification Item History of this document according to AUTOSAR R4.0.3

F.3.1 Changed Constraints in R4.0.3

N/A

F.3.2 Changed Specification Items in R4.0.3

N/A

F.3.3 Added Constraints in R4.0.3

Number	Heading
[constr_3020]	CommmunicationDirection of containedIPduGroups
[constr_3021]	Mapping of `SensorActuatorSwComponents` to SensorActuator `HwElements`
[constr_3024]	Usage of `triggeredWithoutRepetition` and `triggeredOnChangeWithoutRepetition` is not allowed for signal groups and group signals.
[constr_3025]	Usage of NPdus in TpConnections
[constr_3026]	valid `EmptySignalMappings`

Table F.3: Added Constraints in R4.0.3

F.3.4 Added Specification Items in R4.0.3

Number	Heading
[TPS_SYST_01000]	`FlatInstanceDescriptor` roles

Table F.4: Added Specification Items in 4.0.3

F.3.5 Deleted Constraints in R4.0.3

N/A

F.3.6 Deleted Specification Items in R4.0.3

N/A

F.4 Constraint and Specification Item History of this document according to AUTOSAR R4.1.1

F.4.1 Changed Constraints in R4.1.1

Number	Heading
[constr_3018]	Number of FlexRay channels

Table F.5: Changed Constraints in R4.1.1

F.4.2 Changed Specification Items in R4.1.1

N/A

F.4.3 Added Constraints in R4.1.1

Number	Heading
[constr_1198]	TriggerToSignalMapping.systemSignals eligible for a TriggerToSignalMapping
[constr_1199]	ISignals relating to systemSignals eligible for a TriggerToSignalMapping
[constr_1206]	DataMapping to PRPortPrototype
[constr_1207]	Existence of the attribute DataMapping.communicationDirection in the context of a SenderReceiverInterface or TriggerInterface
[constr_1208]	Existence of the attribute DataMapping.communicationDirection in the context of a ClientServerInterface
[constr_1265]	DoIpGidSynchronizationNeeds can only exist once per ECU_EXTRACT
[constr_1266]	DoIpGidNeeds can only exist once per ECU_EXTRACT
[constr_1267]	DoIpActivationLineNeeds can only exist once per ECU_EXTRACT
[constr_3027]	Existence of ecuExtractVersion
[constr_3028]	FibexElements
[constr_3029]	Assign-Frame command usage
[constr_3030]	valid relationship between ECUMapping and EcuInstance
[constr_3031]	Complete System Description does not have outside ports
[constr_3032]	Combinations of SwcToEcuMapping targets
[constr_3033]	Criteria for primitive argument mapping

[constr_3034]	Values of LinSlaveConfig and LinSlave attributes
[constr_3035]	CanNm user data configuration in case NID/CBV are enabled
[constr_3036]	Pdus in CAN and LIN Frames
[constr_3037]	maximum Frame frameLength for CAN and LIN
[constr_3038]	maximum Frame frameLength for FlexRay
[constr_3039]	pncIdentifier range
[constr_3040]	Restriction of pncIdentifier values
[constr_3041]	pncVectorOffset range
[constr_3042]	pncVectorLength range
[constr_3043]	pncVector configuration in AUTOSAR Com
[constr_3044]	CBV configuration in case partial network is used
[constr_3045]	Signal content evaluation vs. Mode evaluation
[constr_3046]	Consistency of TransmissionModeCondition.iSignalInIPdu
[constr_3047]	Uniqueness of macMulticastAddresses
[constr_3048]	Range of vlanIdentifier
[constr_3049]	Role of SystemSignal in inter-ECU client server communication with clients located on different ECUs
[constr_3050]	J1939Cluster uses exactly one CanPhysicalChannel
[constr_3051]	Restriction of ISignalMapping references
[constr_3052]	Complete ISignalMapping of ISignalGroup signals
[constr_3053]	Complete ISignalMapping of target ISignalGroup
[constr_3054]	SystemSignal that is part of exactly one SystemSignalGroup and is not transmitted additionally as standalone SystemSignal in a complete System Description
[constr_3055]	SystemSignalGroup in a complete System Description
[constr_3056]	pduLength of the NmPdu
[constr_3057]	Maximal one BusspecificNmEcu per NmEcu and bus system is allowed to be defined
[constr_3058]	References from SenderRecArrayElementMapping and from SenderRecRecordElementMapping to SystemSignals are not allowed within a SenderReceiverCompositeElementToSignalMapping
[constr_3059]	Mandatory DataMapping on the receiver side for elements of a composite data type
[constr_3060]	Usage of networkRepresentationProps and physicalProps
[constr_3061]	CompuMethod specification in networkRepresentationProps
[constr_3062]	The EcuInstance that is referenced from a specific CouplingElement shall be connected to the same EthernetCluster as the specific CouplingElement
[constr_3063]	Usage of portNumber and dynamicallyAssigned with value “true” is mutually exclusive
[constr_3064]	Usage of serviceInstance, eventHandler and eventGroup references
[constr_3065]	Mapping of queued Triggers to SystemSignals is prohibited
[constr_3066]	Restriction of SenderComSpecs that refer to dataElements mapped to the same SystemSignal
[constr_3067]	initValue defined in the context of ISignal
[constr_3068]	DoIPPowerModeStatusNeeds in the category ECU_EXTRACT
[constr_3501]	Role of SystemSignal in 1:n communication
[constr_3502]	Role of SystemSignal in n:1 sender-receiver communication
[constr_3503]	SystemSignal that is not part of a SystemSignalGroup in a complete System Description
[constr_3505]	Criteria for primitive Data Mapping
[constr_3506]	Mapping of composite data type to SystemSignals in SystemSignalGroup
[constr_3508]	Value of nmReadySleepTime
[constr_3514]	No two ISignalToIPduMappings shall reference the identical ISignal

Table F.6: Added Constraints in R4.1.1

F.4.4 Added Specification Items in R4.1.1

Number	Heading
[TPS_SYST_01001]	Definition of SwcToEcuMapping
[TPS_SYST_01002]	System Category
[TPS_SYST_01003]	Standardized System Category Definitions
[TPS_SYST_01004]	Definition of AUTOSAR ECU
[TPS_SYST_01005]	Definition of EcuInstance
[TPS_SYST_01006]	Assign ECU type to EcuInstance
[TPS_SYST_01007]	Definition of CommunicationController
[TPS_SYST_01008]	Assign CommunicationController to the AUTOSAR Communication Peripheral
[TPS_SYST_01009]	Definition of CommunicationConnector
[TPS_SYST_01010]	Definition of CommunicationCluster
[TPS_SYST_01011]	Definition of PhysicalChannel
[TPS_SYST_01012]	Different Properties of LinMaster and LinSlave
[TPS_SYST_01013]	EcuInstance stands for its own
[TPS_SYST_01014]	Semantics of CommunicationControllerMapping
[TPS_SYST_01015]	Semantics of HwPortMapping
[TPS_SYST_01016]	System Extract, Ecu System Description and Ecu Extract may have ports
[TPS_SYST_01017]	The role of the top-level software composition
[TPS_SYST_01019]	Mapping of topology elements to elements of the ECU Resource Template
[TPS_SYST_01020]	Unconditional mapping of atomic Software Components
[TPS_SYST_01021]	Mapping of CompositionSwComponentType
[TPS_SYST_01022]	Prototype of a ParameterSwComponentType can be mapped to more than one ECU
[TPS_SYST_01023]	Prototype of an ServiceProxySwComponentType can be mapped to more than one ECU
[TPS_SYST_01024]	Component Clustering
[TPS_SYST_01025]	Clustering of Compositions
[TPS_SYST_01026]	Separation of Compositions
[TPS_SYST_01027]	Mapping of specific SW components to dedicated Ecus
[TPS_SYST_01028]	Task of the System Generator
[TPS_SYST_01029]	Mapping of specific SW components to exclusive Ecus
[TPS_SYST_01030]	Representation of VariableDataPrototypes and ClientServerOperations in System Description
[TPS_SYST_01032]	Independence of SystemSignals from CommunicationClusters
[TPS_SYST_01033]	DataMapping and SwConnector
[TPS_SYST_01034]	Data Mappings can be applied to compositions and atomic software components
[TPS_SYST_01035]	Transformation of Data Mappings during flattening
[TPS_SYST_01036]	No additional Data Mappings in composition substructure
[TPS_SYST_01037]	primitive Data Mapping of UINT8-Arrays
[TPS_SYST_01038]	Mapping of primitive arguments
[TPS_SYST_01039]	primitive Argument Mapping of UINT8-Arrays
[TPS_SYST_01040]	Mapping of composite arguments
[TPS_SYST_01041]	CommonSignalPath definition
[TPS_SYST_01042]	ForbiddenSignalPath definition
[TPS_SYST_01043]	PermissibleSignalPath definition
[TPS_SYST_01044]	SeparateSignalPath definition
[TPS_SYST_01045]	Component Seperation
[TPS_SYST_01046]	ShortNames of LinSlaveConfig and LinSlave
[TPS_SYST_01048]	Handling of large IPdus

[TPS_SYST_01049]	Handling of IPdus with dynamic signals
[TPS_SYST_01050]	SystemSignal in the System Extract and ECU Extract
[TPS_SYST_01051]	Handling of DcmIPdus
[TPS_SYST_01052]	Routing of UserDefinedPdus, NmPdus, NPdus
[TPS_SYST_01053]	Low-level routing of NPdus
[TPS_SYST_01054]	Routing of DcmIPdus
[TPS_SYST_01055]	Routing of ISignalIPdus that are part of a MultiplexedIPdu
[TPS_SYST_01056]	Routing of ISignalIPdus, UserDefinedIPdus, MultiplexedIPdus
[TPS_SYST_01057]	Routing of NmPdus
[TPS_SYST_01058]	Pdu Gateway where an Ecu only routes a PduTriggering without being interested in the content
[TPS_SYST_01059]	Relationship between FrameTriggering and CommConnectorPort
[TPS_SYST_01060]	Relationship between PduTriggering and CommConnectorPort
[TPS_SYST_01061]	Relationship between ISignalTriggering and CommConnectorPort
[TPS_SYST_01062]	Network representation of an ISignal
[TPS_SYST_01063]	Context of network representation of an ISignal
[TPS_SYST_01064]	Transmit/Receive Semantics of Pdu Pools
[TPS_SYST_01065]	Mapping onto the of ComSignalType enumeration
[TPS_SYST_01066]	Derivation of Tx COM Signals
[TPS_SYST_01067]	Derivation of Rx COM Signals
[TPS_SYST_01068]	Bit Counting in AUTOSAR
[TPS_SYST_01069]	Bit Order in AUTOSAR
[TPS_SYST_01069]	Bit Order in AUTOSAR
[TPS_SYST_01070]	E2E Protection of ISignalGroups
[TPS_SYST_01071]	E2E Protection of several ISignalGroups in one ISignalIPdu
[TPS_SYST_01072]	Offset attributes of EndToEndDescription
[TPS_SYST_01073]	E2E Protection via COM Callouts
[TPS_SYST_01074]	E2E Protection in the E2E Wrapper
[TPS_SYST_01075]	Signal content evaluation via TransmissionModeCondition
[TPS_SYST_01076]	Mode evaluation via modeDrivenTrueCondition
[TPS_SYST_01077]	Mapping of Com Transmission Modes to System Template elements
[TPS_SYST_01078]	Dynamic Part of a MultiplexedIPdu
[TPS_SYST_01079]	Static Part of a MultiplexedIPdu
[TPS_SYST_01080]	Sending or receiving of a MultiplexedIPdu in System Extract/ECU Extract
[TPS_SYST_01081]	Gatewaying of a MultiplexedIPdu in System Extract/ECU Extract
[TPS_SYST_01082]	Receiving and gatewaying of a MultiplexedIPdu in System Extract/ECU Extract
[TPS_SYST_01083]	A Frame represents a general design object that is used to describe the layout of the included Pdus as a reusable asset.
[TPS_SYST_01084]	FrameTriggering
[TPS_SYST_01085]	Transmission of a Frame multiple times within one communication cycle
[TPS_SYST_01086]	Number of Ethernet channels
[TPS_SYST_01087]	Role of SystemSignal in inter-ECU client server communication with clients located on the same ECU
[TPS_SYST_01088]	NetworkEndpoint priority
[TPS_SYST_01089]	ApplicationEndpoint priority
[TPS_SYST_01090]	valid NetworkEndpoint
[TPS_SYST_01091]	Definition of SoAdConfig
[TPS_SYST_01092]	Transmission of multiple Pdus over the same SocketConnection
[TPS_SYST_01093]	Activation/Deactivation of SoAdRoutingGroups
[TPS_SYST_01094]	allowed key/value CapabilityRecord combinations
[TPS_SYST_01095]	tagged VLANs
[TPS_SYST_01096]	untagged VLANs

[TPS_SYST_01097]	Assignment of CouplingPorts to a VLAN
[TPS_SYST_01098]	Assignment of CouplingPorts to an “untagged” VLAN
[TPS_SYST_01099]	Context of TpConfig
[TPS_SYST_01100]	TP routing using the same transport protocol
[TPS_SYST_01101]	TP routing using different transport protocols
[TPS_SYST_01102]	FlexrayTpConnectionControl reuse
[TPS_SYST_01103]	FlexrayTpConnection shall specify one txPduPool
[TPS_SYST_01104]	FlexrayTpConnection with several receivers
[TPS_SYST_01105]	CanTpConnection with several receivers
[TPS_SYST_01106]	Usage of additional directPdu in case of variable length tpSdu
[TPS_SYST_01107]	Definition of NmCoordinator
[TPS_SYST_01108]	ProvidedServiceInstance priority
[TPS_SYST_01109]	RTE fan-out support
[TPS_SYST_01110]	Com Signal Gateway fan-out support
[TPS_SYST_01111]	Pdu Router fan-out support
[TPS_SYST_01112]	FlexRay dual channel Pdu Router interaction
[TPS_SYST_01113]	FlexRay Interface fan-out support
[TPS_SYST_01114]	Frame fan-out support
[TPS_SYST_01115]	CDD communication support
[TPS_SYST_01116]	Frame Mapping is not supported by the AUTOSAR BSW
[TPS_SYST_01117]	Pdu Gateway support
[TPS_SYST_01118]	Support of Multicast Pdu routing
[TPS_SYST_01119]	Signal Gateway support
[TPS_SYST_01120]	Precedence of ISignalMappings
[TPS_SYST_01121]	Support of Mulicast signal routing
[TPS_SYST_01122]	partial routing between ISignalGroups
[TPS_SYST_01123]	System Extract may cover one or many EcuInstances
[TPS_SYST_01124]	SystemSignal fan-out and fan-in
[TPS_SYST_01125]	SystemSignalGroup fan-out and fan-in
[TPS_SYST_01126]	Resource Consumption for RTE and basic software
[TPS_SYST_01127]	CDD Topology support
[TPS_SYST_01128]	Communication over FlexRay
[TPS_SYST_01129]	Communication over LIN
[TPS_SYST_01130]	Communication over CAN
[TPS_SYST_01131]	TCP/IP and UDP/IP communication over Ethernet
[TPS_SYST_01132]	Communication over SAE J1939
[TPS_SYST_01133]	Partial Network Clusters
[TPS_SYST_01134]	Abstract System Description
[TPS_SYST_01135]	Refactoring of an Abstract System Description into a project specific technical view of the software architecture
[TPS_SYST_01136]	ViewMapSet and ViewMap are used to trace the transformations between different models
[TPS_SYST_01137]	Several DataMappings may be defined for the same SystemSignal
[TPS_SYST_01138]	Low-level routing of XcpPdus
[TPS_SYST_01139]	Ecu Extract covers exactly one EcuInstance
[TPS_SYST_01140]	Ecu Extract contains only SwComponentPrototypes of type AtomicSwComponentType in the RootSwCompositionPrototype
[TPS_SYST_01141]	Derivation of ComSignalType
[TPS_SYST_01142]	Rules for the creation of Triggerings and Ports on the sender side
[TPS_SYST_01143]	DataMapping on the sender side for elements of a composite data type
[TPS_SYST_01144]	Physical properties of a System Signal
[TPS_SYST_01145]	PortInterfaceMappings in the ECU Extract
[TPS_SYST_01146]	Generic CanTpConnections

[TPS_SYST_01147]	Generic J1939TpConnections
[TPS_SYST_01148]	Mapping of IN and INOUT ArgumentDataPrototypes to callSignals
[TPS_SYST_01149]	Mapping of OUT and INOUT ArgumentDataPrototypes to returnSignals
[TPS_SYST_01150]	Mapping of returnSignal and callSignal to COM Signal
[TPS_SYST_01151]	DataMapping reference to an EventHandler
[TPS_SYST_01152]	DataMapping reference to a ConsumedEventGroup
[TPS_SYST_01153]	Atomic transport of SystemSignalGroups
[TPS_SYST_01154]	CAN Controller support of CAN FD frames
[TPS_SYST_03000]	Co-existing System with category SYSTEM_DESCRIPTION and System with category SYSTEM_EXTRACT
[TPS_SYST_05000]	System Description doesn't use a complete Software Component Description
[TPS_SYST_05001]	Send a Trigger across a network
[TPS_SYST_05002]	The value of startPosition is irrelevant

Table F.7: Added Specification Items in 4.1.1

F.4.5 Deleted Constraints in R4.1.1

[constr_3016]	Number of Ethernet channels

Table F.8: Deleted Constraints in R4.1.1

F.4.6 Deleted Specification Items in R4.1.1

N/A

F.5 Constraint and Specification Item History of this document according to AUTOSAR R4.1.2

F.5.1 Changed Specification Items in R4.1.2

Number	Heading
[TPS_SYST_01052]	Routing of UserDefinedPdus, NmPdus, NPdus, GeneralPurposePdus
[TPS_SYST_01056]	Routing of ISignalIPdus, UserDefinedIPdus, MultiplexedIPdus, GeneralPurposeIPdus
[TPS_SYST_01138]	Low-level routing of XcpPdus

Table F.9: Added Specification Items in 4.1.2

F.5.2 Added Specification Items in R4.1.2

Number	Heading
[TPS_SYST_02001]	networkRepresentationProps are mandatory in case the dataTypePolicy is set to override or legacy
[TPS_SYST_02002]	SoAdRoutingGroup for Services with Methods

[TPS_SYST_02003]	SoAdRoutingGroups for Services with event groups
[TPS_SYST_02004]	SoAdRoutingGroups for Services with event groups that contain triggered events
[TPS_SYST_02005]	Low-level routing of J1939DcmIPdus
[TPS_SYST_02006]	Usage of networkRepresentationFromComSpec
[TPS_SYST_02007]	Usage of SocketConnection attributes in the unicast server view
[TPS_SYST_02008]	Usage of SocketConnection attributes in the unicast client view
[TPS_SYST_02009]	Usage of SocketConnection attributes in the multicast server view
[TPS_SYST_02010]	Usage of SocketConnection attributes in the multicast client view

Table F.10: Added Specification Items in 4.1.2

F.5.3 Added Constraints in R4.1.2

Number	Heading
[constr_3069]	Allowed CanNmCluster.nmNidPosition values
[constr_3070]	Allowed CanNmCluster.nmCbvPosition values
[constr_3071]	CanNmCluster.nmCbvPosition and CanNmCluster.nmNidPosition shall never have the same value
[constr_3073]	nmVoteInformation only valid for FrNm
[constr_3074]	No TransmissionAcknowledgementRequest for multiple senders
[constr_3078]	Allowed UdpNmCluster.nmNidPosition values
[constr_3079]	Allowed UdpNmCluster.nmCbvPosition values
[constr_3080]	UdpNmCluster.nmCbvPosition and UdpNmCluster.nmNidPosition shall never have the same value
[constr_3081]	Value of category in GeneralPurposePdu
[constr_3082]	Value of category in GeneralPurposeIPdu
[constr_3083]	Exactly one AtomicSwComponentType on an EcuInstance may use General-CallbackEventDataChanged / GeneralCallbackEventStatusChange
[constr_3084]	Service port in the role PowerTakeOff
[constr_3085]	Service port in the role CallbackDCMRequestServices

Table F.11: Added Constraints in R4.1.2

F.5.4 Changed Constraints in R4.1.2

Number	Heading
[constr_2025]	Uniqueness of symbol attributes

Table F.12: Changed Constraints in R4.1.2

F.5.5 Deleted Constraints in R4.1.2

[constr_3066]	Restriction of SenderComSpecs that refer to dataElements mapped to the same SystemSignal

Table F.13: Deleted Constraints in R4.1.2

F.6 Constraint and Specification Item History of this document according to AUTOSAR R4.1.3

F.6.1 Changed Specification Items in R4.1.3

N/A

F.6.2 Added Specification Items in R4.1.3

Number	Heading
[TPS_SYST_01155]	Routing of ISignalGroups
[TPS_SYST_01156]	Definition of ISignalTriggerings is allowed for ISignalGroups and for GroupSignals
[TPS_SYST_01157]	Allowed usage of attributes for ISignals, ISignalGroups and GroupSignals
[TPS_SYST_02011]	initValues of receivers that are mapped to the same Ecu
[TPS_SYST_02012]	initValue and invalidValue represent internal values

Table F.14: Added Specification Items in 4.1.3

F.6.3 Deleted Specification Items in R4.1.3

Number	Heading
[TPS_SYST_01124]	SystemSignal fan-out and fan-in
[TPS_SYST_01125]	SystemSignalGroup fan-out and fan-in

Table F.15: Deleted Specification Items in 4.1.3

F.6.4 Added Constraints in R4.1.3

[constr_3086]	Role of SystemSignal in n:1 sender-receiver communication
[constr_3087]	DataMapping to PRPortPrototype
[constr_3088]	SystemSignal that is not part of a SystemSignalGroup in a complete System Description
[constr_3089]	SystemSignal that is part of exactly one SystemSignalGroup and is not transmitted additionally as standalone SystemSignal in a complete System Description
[constr_3090]	TpSdu transmission on a PhysicalChannel
[constr_3094]	Consistent ISignalPort.communicationDirection for ISignalTriggerings of ISignalGroups and contained ISignals

Table F.16: Added Constraints in R4.1.3

F.6.5 Changed Constraints in R4.1.3

[constr_3051]	Restriction of ISignalMapping references

Table F.17: Changed Constraints in R4.1.3

F.6.6 Deleted Constraints in R4.1.3

[constr_3502]	Role of SystemSignal in n:1 sender-receiver communication
[constr_1206]	DataMapping to PRPortPrototype
[constr_3503]	SystemSignal that is not part of a SystemSignalGroup in a complete System Description
[constr_3054]	SystemSignal that is part of exactly one SystemSignalGroup and is not transmitted additionally as standalone SystemSignal in a complete System Description

Table F.18: Deleted Constraints in R4.1.3

G Mentioned Class Tables

For the sake of completeness, this chapter contains a set of class tables representing meta-classes mentioned in the context of this document but which are not contained directly in the scope of describing specific meta-model semantics.

Class	**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,Packageable Element,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table G.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,Collectable Element,Identifiable,MultilanguageReferrable,Referrable			
Attribute	***Datatype***	***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, variation Point.shortLabel vh.latestBindingTime=blueprintDerivationTime xml.sequenceOffset=30
element	PackageableEle ment	*	aggr	Elements that are part of this package **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variation Point.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=20

Attribute	Datatype	Mul.	Kind	Note
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 G.2: ARPackage

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			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
providedComSpec	PPortComSpec	*	aggr	Provided communication attributes per interface element (data element or operation).

Table G.3: AbstractProvidedPortPrototype

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	**Datatype**	**Mul.**	**Kind**	**Note**
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 G.4: ApplicationArrayDataType

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			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**

Attribute	Datatype	Mul.	Kind	Note
–	–	–	–	–

Table G.5: ApplicationCompositeDataType

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 instanceRefs.			
Base	ARObject,AtpFeature,AtpPrototype,DataPrototype,Identifiable,MultilanguageReferrable,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
type	ApplicationDataType	1	tref	This represents the corresponding data type. **Stereotypes:** isOfType

Table G.6: ApplicationCompositeElementDataPrototype

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 ApplicationDataTypes only.			
Base	ARElement,ARObject,AtpBlueprint,AtpBlueprintable,AtpClassifier,AtpType,AutosarDataType,CollectableElement,Identifiable,MultilanguageReferrable,PackageableElement,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
–	–	–	–	–

Table G.7: ApplicationDataType

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	**Datatype**	**Mul.**	**Kind**	**Note**
–	–	–	–	–

Table G.8: ApplicationPrimitiveDataType

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	Datatype	Mul.	Kind	Note
–	–	–	–	–

Table G.9: ApplicationRecordElement

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,MultilanguageReferrable,Referrable			
Attribute	Datatype	Mul.	Kind	Note
direction	ArgumentDirectionEnum	1	attr	This attribute specifies the direction of the argument prototype.
serverArgumentImplPolicy	ServerArgumentImplPolicyEnum	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 semantic as if the attribute is set to useArgumentType
typeBlueprint	AutosarDataType	0..1	ref	This allows to denote the intended type within blueprints. It shall be replaced by a proper type when deriving Interfaces from the Blueprint. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=blueprintDerivationTime

Table G.10: ArgumentDataPrototype

Enumeration	ArgumentDirectionEnum
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::PrimitiveTypes
Note	Use cases: • Arguments in ClientServerOperation can have different directions that need to be formally indicated because they have an impact on how the function signature looks like eventually. • Arguments in BswModuleEntry already determine a function signature, but the direction is used to specify the semantics, especially of pointer arguments.
Literal	Description
in	The argument value is passed to the callee.

inout	The argument value is passed to the callee but also passed back from the callee to the caller.
out	The argument value is passed from the callee to the caller.

Table G.11: ArgumentDirectionEnum

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	*Datatype*	*Mul.*	*Kind*	*Note*
provider	AbstractProvidedPortPrototype	0..1	iref	Instance of providing port.
requester	AbstractRequiredPortPrototype	0..1	iref	Instance of requiring port.

Table G.12: AssemblySwConnector

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			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
internalBehavior	SwcInternalBehavior	0..1	aggr	The SwcInternalBehaviors owned by an AtomicSwComponentType can be located in a different physical file. Therefore the aggregation is «atpSplitable». **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=internalBehavior, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
symbolProps	SymbolProps	0..1	aggr	This represents the SymbolProps for the AtomicSwComponentType. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=shortName

Table G.13: AtomicSwComponentType

Class	AtpInstanceRef (abstract)			
Package	M2::AUTOSARTemplates::GenericStructure::AbstractStructure			
Note	An M0 instance of a classifier may be represented as a tree rooted at that instance, where under each node come the sub-trees representing the instances which act as features under that node. An instance ref specifies a navigation path from any M0 tree-instance of the base (which is a classifier) to a leaf (which is an instance of the target).			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
atpBase	AtpClassifier	1	ref	This is the base from which the navigaion path starts. **Stereotypes:** atpAbstract; atpDerived
atpContextElement (ordered)	AtpPrototype	*	ref	This is one particular step in the navigation path. **Stereotypes:** atpAbstract
atpTarget	AtpFeature	1	ref	This is the target of the instance ref. In other words it is the terminal of the navigation path. **Stereotypes:** atpAbstract

Table G.14: AtpInstanceRef

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			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
type	AutosarDataType	1	tref	This represents the corresponding data type. **Stereotypes:** isOfType

Table G.15: 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			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
swDataDefProps	SwDataDefProps	0..1	aggr	The properties of this AutosarDataType.

Table G.16: AutosarDataType

Class	**BaseType (abstract)**			
Package	M2::AUTOSARTemplates::CommonStructure::BaseTypes			
Note	This abstract meta-class represents the ability to specify a platform dependant base type.			
Base	ARElement,ARObject,CollectableElement,Identifiable,Multilanguage Referrable,PackageableElement,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
baseType Definition	BaseTypeDefini tion	1	aggr	This is the actual definition of the base type. **Tags:** xml.roleElement=false; xml.roleWrapper Element=false; xml.sequenceOffset=20; xml.type Element=false; xml.typeWrapperElement=false

Table G.17: BaseType

Class	**BaseTypeDirectDefinition**			
Package	M2::AUTOSARTemplates::CommonStructure::BaseTypes			
Note	This BaseType is defined directly (as opposite to a derived BaseType)			
Base	ARObject,BaseTypeDefinition			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
baseType Encoding	BaseTypeEnco dingString	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
baseType Size	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
maxBaseT ypeSize	PositiveInteger	0..1	attr	Describes the maximum length of the BaseType in bits. **Tags:** xml.sequenceOffset=80
memAlign ment	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

Attribute	Datatype	Mul.	Kind	Note

nativeDeclaration	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 ImplementationDataTypes will not be generated as a typedef by RTE.
				
				If a nativeDeclaration type is given it shall fulfill the characteristic given by basetypeEncoding and baseTypeSize.
				
				This is required to ensure the consistent handling and interpretation by software components, RTE, COM and MCM systems.
				
				Tags: xml.sequenceOffset=120

Table G.18: BaseTypeDirectDefinition

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,Internal Behavior,MultilanguageReferrable,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
distinguishedPartition	BswDistinguishedPartition	*	aggr	Indicates an abstract partition context in which the enclosing BswModuleEntity can be executed. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=60
entity	BswModuleEntity	1..*	aggr	A code entity for which the behavior is described **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=5

Attribute	Datatype	Mul.	Kind	Note
event	BswEvent	*	aggr	An event required by this module behavior. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=10
internalTriggeringPoint	BswInternalTriggeringPoint	*	aggr	An internal triggering point. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=2
modeReceiverPolicy	BswModeReceiverPolicy	*	aggr	Implementation policy for the reception of mode switches. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=25
modeSenderPolicy	BswModeSenderPolicy	*	aggr	Implementation policy for providing a mode group. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=20
perInstanceParameter	ParameterDataPrototype	*	aggr	Describes a read only memory object containing characteristic value(s) needed by this BswInternalBehavior. The role name perInstanceParameter is chosen in analogy to the similar role in the context of SwcInternalBehavior. 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:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=45
receptionPolicy	BswDataReceptionPolicy	*	aggr	Data reception policy for inter-partition and/or inter-core communication. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=55
schedulerNamePrefix	BswSchedulerNamePrefix	*	aggr	Optional definition of one or more prefixes to be used for the BswScheduler. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=50

Attribute	Datatype	Mul.	Kind	Note
serviceDependency	BswServiceDependency	*	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, variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=40
triggerDirectImplementation	BswTriggerDirectImplementation	*	aggr	Specifies a trigger to be directly implemented via OS calls. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=15

Table G.19: BswInternalBehavior

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	**Datatype**	**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 G.20: ClientServerInterface

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	*Datatype*	*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 G.21: ClientServerOperation

Enumeration	**CommunicationDirectionType**
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication
Note	Describes the communication direction.
Literal	*Description*
in	Reception (Input)
out	Transmission (Output)

Table G.22: CommunicationDirectionType

Class	**CompositionSwComponentType**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Composition			
Note	A CompositionSwComponentType aggregates SwComponentPrototypes (that in turn are typed by SwComponentTypes) 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	*Datatype*	*Mul.*	*Kind*	*Note*

Attribute	Datatype	Mul.	Kind	Note
component	SwComponentPrototype	*	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 SwComponentPrototypes 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 AssemblySwConnectors between ApplicationSwComponentTypes and ServiceSwComponentTypes during the ECU integration. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=postBuild
constantValueMapping	ConstantSpecificationMappingSet	*	ref	Reference to the ConstantSpecificationMapping to be applied for initValues of PPortComSpecs and RPortComSpec.

Attribute	Datatype	Mul.	Kind	Note
dataTypeMapping	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 CompositionSwComponentTypes. Rationale: if the mapping of ApplicationDataTypes on the delegated and inner PortPrototype matches then the mapping to ImplementationDataTypes is not impacting compatibility.
instantiationRTEEventProps	InstantiationRTEEventProps	*	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 G.23: CompositionSwComponentType

Class	CompuMethod			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::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	**Datatype**	**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

Attribute	Datatype	Mul.	Kind	Note
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 G.24: CompuMethod

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	***Datatype***	***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 G.25: ConditionByFormula

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			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
swDataDefProps	SwDataDefProps	0..1	aggr	This property allows to specify data definition properties which apply on data prototype level.

Table G.26: DataPrototype

Class	DataTypeMap			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::Datatypes			
Note	This class represents the relationship between ApplicationDataType and its implementing ImplementationDataType.			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
applicationDataType	ApplicationDataType	1	ref	This is the corresponding ApplicationDataType
implementationDataType	ImplementationDataType	1	ref	This is the corresponding ImplementationDataType.

Table G.27: 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	**Datatype**	**Mul.**	**Kind**	**Note**
dataTypeMap	DataTypeMap	*	aggr	This is one particular association between an ApplicationDataType and its ImplementationDataType.
modeRequestTypeMap	ModeRequestTypeMap	*	aggr	This is one particular association between an ModeDeclarationGroup and its ImplementationDataType.

Table G.28: DataTypeMappingSet

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	**Datatype**	**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 CompositionType

Table G.29: DelegationSwConnector

Class	Describable (abstract)			
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::Identifiable			
Note	This meta-class represents the ability to add a descriptive documentation to non identifiable elements.			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
desc	MultiLanguage OverviewParagr aph	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 particlar how the object is built or used) should go to "introduction". **Tags:** xml.sequenceOffset=-60
category	CategoryString	0..1	attr	This element assigns a category to the parent element. The category is intended to specialize the usage and/or the content identifiable object. Such a specialization may also impose particular semantic constraints on the entire substructure (not only the identifiable itself). **Tags:** xml.sequenceOffset=-50
adminData	AdminData	0..1	aggr	This represents the administrative data for the describable object. **Tags:** xml.sequenceOffset=-20
introductio n	Documentation Block	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

Table G.30: Describable

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,Service Needs			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
–	–	–	–	–

Table G.31: DoIpPowerModeStatusNeeds

Class	DoIpActivationLineNeeds			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	A DoIP entity needs to be informed when an external tester is attached or activated. The DoIpActivationServiceNeeds 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	Datatype	Mul.	Kind	Note
–	–	–	–	–

Table G.32: 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	Datatype	Mul.	Kind	Note
–	–	–	–	–

Table G.33: 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	Datatype	Mul.	Kind	Note
–	–	–	–	–

Table G.34: 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,Service Needs			
Attribute	Datatype	Mul.	Kind	Note
–	–	–	–	–

Table G.35: 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			
Attribute	Datatype	Mul.	Kind	Note
–	–	–	–	–

Table G.36: DoIpServiceNeeds

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,Multilanguage Referrable,PackageableElement,Referrable			
Attribute	Datatype	Mul.	Kind	Note
associated ComIPduGroup	ISignalIPduGroup	*	ref	With this reference it is possible to identify which ISignalIPduGroups are applicable for which CommunicationConnector/ ECU. Only top level ISignalIPduGroups shall be referenced by an EcuInstance. If an ISignalIPduGroup contains other ISignalIPduGroups than these contained ISignalIPduGroups shall not be referenced by the EcuInstance. Contained ISignalIPduGroups are associated to an EcuInstance via the top level ISignalIPduGroup.
associated PdurIPduGroup	PdurIPduGroup	*	ref	With this reference it is possible to identify which PduR IPdu Groups are applicable for which CommunicationConnector/ ECU.

Attribute	Datatype	Mul.	Kind	Note
canTpAddress	CanTpAddress	*	ref	Please note that this reference is deprecated and will be removed in future. A Tp Address can be assigned to an ECU without an existing TP Configuration. If TpNodes are described this reference shall not be used. **Tags:** atp.Status=obsolete; atp.StatusRevisionBegin=4.1.3
comConfigurationGwTimeBase	TimeValue	0..1	attr	The period between successive calls to Com_MainFunctionRouteSignals of the AUTOSAR COM module in seconds.
comConfigurationRxTimeBase	TimeValue	0..1	attr	The period between successive calls to Com_MainFunctionRx of the AUTOSAR COM module in seconds.
comConfigurationTxTimeBase	TimeValue	0..1	attr	The period between successive calls to Com_MainFunctionTx of the AUTOSAR COM module in seconds.
comEnableMDTForCyclicTransmission	Boolean	0..1	attr	Enables for the Com module of this EcuInstance the minimum delay time monitoring for cyclic and repeated transmissions (TransmissionModeTiming has cyclicTiming assigned or eventControlledTiming with numberOfRepetitions > 0).
commController	CommunicationController	1..*	aggr	CommunicationControllers of the ECU.
connector	CommunicationConnector	*	aggr	All channels controlled by a single controller.
diagnosticAddress	Integer	0..1	attr	An ECU specific ID for responses of diagnostic routines.
partition	EcuPartition	*	aggr	Optional definition of Partitions within an Ecu.
sleepModeSupported	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.
tpAddress	TpAddress	*	ref	Please note that this reference is deprecated and will be removed in future. A Tp Address can be assigned to an ECU without an existing TP Configuration. If TpNodes are described this reference shall not be used. **Tags:** atp.Status=obsolete; atp.StatusRevisionBegin=4.1.3

Attribute	Datatype	Mul.	Kind	Note
wakeUpOverBusSupported	Boolean	1	attr	Driver support for wakeup over Bus.

Table G.37: EcuInstance

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	Datatype	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	ref	This serves as part of the split key in case of more than one EndToEndProtectionVariablePrototype is aggregated in the bound model.

Table G.38: EndToEndProtectionVariablePrototype

Class	ExecutableEntity (abstract)			
Package	M2::AUTOSARTemplates::CommonStructure::InternalBehavior			
Note	Abstraction of executable code.			
Base	ARObject,Identifiable,MultilanguageReferrable,Referrable			
Attribute	Datatype	Mul.	Kind	Note
activationReason	ExecutableEntityActivationReason	*	aggr	If the ExecutableEntity provides at least one activationReason 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.
canEnterExclusiveArea	ExclusiveArea	*	ref	This means that the executable entity can enter/leave the referenced exclusive area through explicit API calls.
exclusiveAreaNestingOrder	ExclusiveAreaNestingOrder	*	ref	This represents the set of ExclusiveAreaNestingOrders recognized by this ExecutableEntity.

Attribute	Datatype	Mul.	Kind	Note
minimumStartInterval	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 ReentrancyLevelEnum for details. Please note that nonReentrant interfaces can have also reentrant or multicoreReentrant implementations, and reentrant interfaces can also have multicoreReentrant implementations.
runsInsideExclusiveArea	ExclusiveArea	*	ref	The executable entity runs completely inside the referenced exclusive area.
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 G.39: ExecutableEntity

Class	ExecutionTime (abstract)			
Package	M2::AUTOSARTemplates::CommonStructure::ResourceConsumption::ExecutionTime			
Note	Base class for several means how to describe the ExecutionTime of software. The required context information is provided through this class.			
Base	ARObject,Identifiable,MultilanguageReferrable,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
exclusiveArea	ExclusiveArea	0..1	ref	Reference to the ExclusiveArea this execution time is provided for.
executableEntity	ExecutableEntity	0..1	ref	The executable entity for which this execution time is described.
hardwareConfiguration	HardwareConfiguration	1	aggr	Provides information on the HardwareConfiguration used to specify this ExecutionTime.
hwElement	HwElement	0..1	ref	The hardware element (e.g. type of ECU) for which the execution time is specified.
includedLibrary	DependencyOnArtifact	*	ref	If this dependency is specified, the execution time of the library code is included in the execution time data for the runnable.
memorySectionLocation	MemorySectionLocation	*	aggr	Provides information on the MemorySectionLocation which is involved in the ExecutionTime description.
softwareContext	SoftwareContext	1	aggr	Provides information on the detailed SoftwareContext used to provide the ExecutionTime description.

Table G.40: ExecutionTime

Class	**FibexElement (abstract)**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore			
Note	ASAM FIBEX elements specifying Communication and Topology.			
Base	ARObject,CollectableElement,Identifiable,MultilanguageReferrable,Packageable Element,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
–	–	–	–	–

Table G.41: FibexElement

Class	**FlexrayCommunicationConnector**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Flexray::FlexrayTopology			
Note	FlexRay specific attributes to the CommunicationConnector			
Base	ARObject,CommunicationConnector,Identifiable,MultilanguageReferrable,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
nmReadySleepTime	Float	0..1	attr	The value of this attribute influences the shutdown behavior of the FlexRay NM. FrNm switches to bus sleep mode nmReadySleepTime seconds after the completion of the last repetition cycle containing a NM vote.
pncFilterDataMask	PositiveUnlimitedInteger	0..1	attr	Bit mask for FlexRay Payload used to configure the FlexRay Transceiver for partial network wakeup. This attribute should not be computed from the pncIdentifier values in order to support future introduction of additional PNCs. Note that for one EcuInstance all contributing pncFilterDataMask will be bitwise ORed to obtain the value of FrNmPnFilterMaskByte. Note that this data mask is calculated over the whole payload (8 Byte) of the NmPdu ignoring the leading bytes which do not contain pncVector information. The number of leading bytes which shall be ignored is equivalent to the value of System.pncVectorOffset.
wakeUpChannel	Boolean	1	attr	Referenced channel used by the node to send a wakeup pattern. (pWakeupChannel)

Table G.42: FlexrayCommunicationConnector

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,Fibex Element,Identifiable,MultilanguageReferrable,PackageableElement,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**

Attribute	*Datatype*	*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 = cSamplesPerBit * gdSampleClockPeriod. Unit: seconds (gdBit)
casRxLow Max	Integer	1	attr	Upper limit of the Collision Avoidance Symbol (CAS) acceptance window. Unit:bitDuration
coldStartAt tempts	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
cycleCount Max	Integer	1	attr	Maximum cycle counter value in a given cluster. Remark: Set to 63 for FlexRay Protocol 2.1 Rev. A compliance.
detectNitEr ror	Boolean	1	attr	Indicates whether NIT error status of each cluster shall be detected or not.
dynamicSl otIdlePhas e	Integer	1	attr	The duration of the dynamic slot idle phase in minislots.
ignoreAfter Tx	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
macroPer Cycle	Integer	1	attr	The number of macroticks in a communication cycle
macrotickD uration	TimeValue	1	attr	Duration of the cluster wide nominal macrotick, expressed in s.
maxWithou tClockCorr ectionFatal	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.
maxWithou tClockCorr ectionPass ive	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.
minislotAct ionPointOff set	Integer	1	attr	The Offset of the action point within a minislot. Unit: macroticks
minislotDur ation	Integer	1	attr	The duration of a minislot (dynamic segment). Unit: macroticks.
networkIdl eTime	Integer	1	attr	The duration of the network idle time in macroticks

Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
networkManagementVectorLength	Integer	1	attr	Length of the Network Management vector in a cluster [bytes]
numberOfMinislots	Integer	1	attr	Number of Minislots in the dynamic segment.
numberOfStaticSlots	Integer	1	attr	The number of static slots in the static segment.
offsetCorrectionStart	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
payloadLengthStatic	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.
sampleClockPeriod	TimeValue	0..1	attr	Sample clock period. Unit: seconds
staticSlotDuration	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
symbolWindowActionPointOffset	Integer	1	attr	Number of macroticks the action point offset is from the beginning of the symbol window [Macroticks].
syncFrameIdCountMax	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.
tranceiverStandbyDelay	Float	0..1	attr	The duration of timer t_TrcvStdbyDelay in seconds. The granularity of this parameter shall be restricted to full FlexRay 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.
transmissionStartSequenceDuration	Integer	1	attr	Number of bits in the Transmission Start Sequence [gdBits].
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.

Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
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.
wakeupRxWindow	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 G.43: FlexrayCluster

Class	**FlexrayTpConfig**			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	This element defines exactly one FlexRay ISO TP Configuration. One FlexRayTpConfig element shall be created for each FlexRay Network in the System that uses FlexRay Iso Tp. **Tags:** atp.recommendedPackage=TpConfigs			
Base	ARObject,CollectableElement,FibexElement,Identifiable,MultilanguageReferrable,PackageableElement,Referrable,TpConfig			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
pduPool	FlexrayTpPduPool	1..*	aggr	Configuration of FlexRay TP Pdu Pools. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
tpAddress	TpAddress	1..*	aggr	Collection of TpAddresses. atpVariation: Derived, because EcuInstance can vary. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
tpConnection	FlexrayTpConnection	*	aggr	Configuration of FlexRay TP Connections. atpVariation: Derived, because TpNode can vary. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
tpConnectionControl	FlexrayTpConnectionControl	*	aggr	Configuration of FlexRay TP Connection Controls. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild

Attribute	Datatype	Mul.	Kind	Note
tpEcu	FlexrayTpEcu	1..*	aggr	Collection of TP Ecus atpVariation: Derived, because EcuInstance can vary. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
tpNode	FlexrayTpNode	*	aggr	Senders and receivers of FlexRay TP messages. atpVariation: Derived, because EcuInstance can vary. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild

Table G.44: FlexrayTpConfig

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			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
bandwidthLimitation	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 (FrTpRxPduPool) which are defined for receiving data via this particular connection. The following constraint is valid only for the System Extract/ECU Extract: In case this connection is applied to the transmitter the rxPduPool holds the actually received NPdus. In case this connection is applied to the receiver the rxPduPool holds the actually sent NPdus.
tpConnectionControl	FlexrayTpConnectionControl	1	ref	Reference to the connection control.

Attribute	Datatype	Mul.	Kind	Note
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 (FrTpTxPduPool) which are defined for sending data via this particular connection. The following constraint is valid only for the System Extract/ECU Extract: In case this connection is applied to the transmitter the txPduPool holds the actually sent NPdus. In case this connection is applied to the receiver the txPduPool holds the actually received NPdus.

Table G.45: FlexrayTpConnection

Class	FlexrayTpConnectionControl			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	Configuration parameters to control a FlexRay TP connection.			
Base	ARObject,Identifiable,MultilanguageReferrable,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
ackType	TpAckType	0..1	attr	This parameter defines the type of acknowledgement which is used for the specific channel.
maxAr	Integer	0..1	attr	This parameter defines the maximum number of trying to send a frame when a TIMEOUT AR occurs (depending on whether retry is configured).
maxAs	Integer	0..1	attr	This parameter defines the maximum number of trying to send a frame when a TIMEOUT AS occurs (depending on whether retry is configured)
maxBufferSize	Integer	0..1	attr	This parameter is only relevant when having retry activated. It limits the maximal buffer size the FrTp can choose in order to limit the amount of Tx buffer that will be requested at the sender side in a segmented transfer.
maxFcWait	Integer	0..1	attr	This attribute defines the maximum number of FlowControl N-PDUs with FlowState "WAIT".
maxFrIf	Integer	0..1	attr	This parameter defines the maximum number of trying to send a frame when the FrIf returns an error
maxNumberOfNpduPerCycle	Integer	0..1	attr	This parameter limits the number of N-Pdus the sender is allowed to transmit within a FlexRay cycle.
maxRetries	Integer	0..1	attr	This parameter defines the maximum number of retries (if retry is configured for the particular channel).
separationCycleExponent	Integer	0..1	attr	Exponent to calculate the minimum number of "Separation Cycles" the sender has to wait for the next transmission of an FrTp N-Pdu.
timeBr	TimeValue	0..1	attr	Time (in seconds) until transmission of the next FlowControl N-PDU.

Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
timeBuffer	TimeValue	0..1	attr	This parameter defines the time of waiting for the next try to get a Tx or Rx buffer. This parameter is equivalent to the temporal distance between two FC.WT N-Pdus in case the buffer request returns busy. Specified in seconds.
timeFrIf	TimeValue	0..1	attr	This parameter defines the time of waiting for the next try to send. Specified in seconds.
timeoutAr	TimeValue	0..1	attr	This parameter states the timeout between the PDU transmit request of the Transport Layer to the FlexRay Interface and the corresponding confirmation of the FlexRay Interface on the receiver side (for FC or AF). Specified in seconds.
timeoutAs	TimeValue	0..1	attr	This attribute states the timeout between the PDU transmit request for the first PDU of the group used in the current connection of the Transport Layer to the FlexRay Interface and the corresponding confirmation of the FlexRay Interface (when having sent the last PDU of the group used in this connection) on the sender side (SF-x, FF-x, CF or FC (in case of Transmit Cancellation)). Specified in seconds.
timeoutBs	TimeValue	0..1	attr	This parameter defines the timeout in seconds for waiting for an FC or AF on the sender side in a 1:1 connection.
timeoutCr	TimeValue	0..1	attr	This parameter defines the timeout value in seconds for waiting for a CF or FF-x (in case of retry) after receiving the last CF or after sending an FC or AF on the receiver side. Specified in seconds.

Table G.46: FlexrayTpConnectionControl

Class	**FlexrayTpNode**			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	TP Node (Sender or Receiver) provides the TP Address and the connection to the Topology description.			
Base	ARObject,Identifiable,MultilanguageReferrable,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
connector	Communication Connector	*	ref	Asscociation to one or more physical connectors (max number of connectors for FlexRay: 2). In a System Description this reference is mandatory. In an ECU Extract this reference is optional (references to ECUs that are not part of the ECU Extract shall be avoided).

Attribute	Datatype	Mul.	Kind	Note
tpAddress	TpAddress	0..1	ref	Reference to the TP Address that is used by the TpNode. This reference is optional in case that the multicast TP Address is used (reference from TpConnection).

Table G.47: FlexrayTpNode

Class	FlexrayTpPduPool			
Package	M2::AUTOSARTemplates::SystemTemplate::TransportProtocols			
Note	FlexrayTpPduPool is a set of N-PDUs which are defined for FrTp sending or receiving purpose.			
Base	ARObject,Identifiable,MultilanguageReferrable,Referrable			
Attribute	Datatype	Mul.	Kind	Note
nPdu	NPdu	1..*	ref	Reference to NPdus that are part of the PduPool.

Table G.48: FlexrayTpPduPool

Class	HeapUsage (abstract)			
Package	M2::AUTOSARTemplates::CommonStructure::ResourceConsumption::HeapUsage			
Note	Describes the heap memory usage of a SW-Component.			
Base	ARObject,Identifiable,MultilanguageReferrable,Referrable			
Attribute	Datatype	Mul.	Kind	Note
hardwareConfiguration	HardwareConfiguration	0..1	aggr	Contains information about the hardware context this heap usage is describing.
hwElement	HwElement	0..1	ref	Specifies for which hardware element (e.g. ECU) this heap usage usage is given.
softwareContext	SoftwareContext	0..1	aggr	Contains details about the software context this heap usage is provided for.

Table G.49: HeapUsage

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	Datatype	Mul.	Kind	Note

Attribute	Datatype	Mul.	Kind	Note
hwElementConnection	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
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 G.50: HwElement

Class	HwPinGroup			
Package	M2::AUTOSARTemplates::EcuResourceTemplate			
Note	This meta-class represents the ability to describe groups of pins which are used to connect hardware elements. This group acts as a bundle of pins. Thereby they allow to describe high level connections. Pin groups can even be nested.			
Base	ARObject,HwDescriptionEntity,Identifiable,MultilanguageReferrable,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
hwPinGroupContent	HwPinGroupContent	1	aggr	This aggregation describes the contained pins/pin groups.

Table G.51: HwPinGroup

Class	HwPortMapping			
Package	M2::AUTOSARTemplates::SystemTemplate::ECUResourceMapping			
Note	HWPortMapping specifies the hwCommunicationPort (defined in the ECU Resource Template) to realize the specified CommunicationConnector in a physical topology.			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
communicationConnector	CommunicationConnector	1	ref	Reference to the CommunicationConnector in the System Template

Attribute	Datatype	Mul.	Kind	Note
hwCommunicationPort	HwPinGroup	1	ref	Reference to the HwPinPortGroup of category CommunicationPort. The connection to the HwCommunicationController is described in the Ecu Resource Description.

Table G.52: HwPortMapping

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			
Attribute	Datatype	Mul.	Kind	Note
–	–	–	–	–

Table G.53: IPdu

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			
Attribute	Datatype	Mul.	Kind	Note
desc	MultiLanguageOverviewParagraph	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
category	CategoryString	0..1	attr	This element assigns a category to the parent element. The category is intended to specialize the usage and/or the content identifiable object. Such a specialization may also impose particular semantic constraints on the entire substructure (not only the identifiable itself). **Tags:** xml.sequenceOffset=-50
adminData	AdminData	0..1	aggr	This represents the administrative data for the identifiable object. **Tags:** xml.sequenceOffset=-40

Attribute	Datatype	Mul.	Kind	Note
annotation	Annotation	*	aggr	Possibility to provide additional notes while defining a model element (e.g. the ECU Configuration Parameter Values). These are not intended as documentation but are mere design notes. **Tags:** xml.sequenceOffset=-25
introductio n	Documentation Block	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". **Tags:** xml.attribute=true

Table G.54: Identifiable

Class	**Implementation (abstract)**			
Package	M2::AUTOSARTemplates::CommonStructure::Implementation			
Note	Description of an implementation a single software component or module.			
Base	ARElement,ARObject,CollectableElement,Identifiable,Multilanguage Referrable,PackageableElement,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
buildAction Manifest	BuildActionMani fest	0..1	ref	A manifest specifying the intended build actions for the software delivered with this implementation. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=codeGenerationTime
codeDescri ptor	Code	1..*	aggr	Specifies the provided implementation code.
compiler	Compiler	*	aggr	Specifies the compiler for which this implementation has been released

Attribute	Datatype	Mul.	Kind	Note
generatedArtifact	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
programmingLanguage	ProgramminglanguageEnum	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
requiredGeneratorTool	DependencyOnArtifact	*	aggr	Relates this Implementation to a generator tool in order to generate additional artifacts during integration. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
resourceConsumption	ResourceConsumption	1	aggr	All static and dynamic resources for each implementation are described within the ResourceConsumption class.
swVersion	RevisionLabelString	1	attr	Software version of this implementation. The numbering contains three levels (like major, minor, patch), its values are vendor specific.

Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
swcBswMapping	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 BswImplementtion or for both.
usedCodeGenerator	String	0..1	attr	Optional: code generator used.
vendorId	PositiveInteger	1	attr	Vendor ID of this Implementation according to the AUTOSAR vendor list

Table G.55: 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,AtpBlueprint,AtpBlueprintable,AtpClassifier,AtpType,AutosarDataType,CollectableElement,Identifiable,MultilanguageReferrable,PackageableElement,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
subElement (ordered)	ImplementationDataTypeElement	*	aggr	Specifies an element of an arrray, 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 ImplementationDataType representing a structure. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
symbolProps	SymbolProps	0..1	aggr	This represents the SymbolProps for the ImplementationDataType. **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 G.56: 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,Identifiable,MultilanguageReferrable,Referrable			
Attribute	**Datatype**	**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 ImplementationDataTypeElement represents the type of each single array element. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
arraySizeSemantics	ArraySizeSemanticsEnum	0..1	attr	This attribute controls the meaning of the value of the array size.
subElement	ImplementationDataTypeElement	*	aggr	Element of an array, struct, or union in case of a nested declaration (i.e. without using "typedefs"). The aggregation of ImplementationDataTypeElement is subject to variability with the purpose to support the conditional existence of elements inside a ImplementationDataType representing a structure. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
swDataDefProps	SwDataDefProps	0..1	aggr	The properties of this ImplementationDataTypeElementt.

Table G.57: ImplementationDataTypeElement

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			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**

Attribute	Datatype	Mul.	Kind	Note
constantMemory	ParameterDataPrototype	*	aggr	Describes a read only memory object containing characteristic value(s) implemented by this InternalBehavior. The shortName of ParameterElementPrototype has to be equal to the "C' identifier of the described constant. The characteristic value(s) might be shared between SwComponentPrototypes of the same SwComponentType. The aggregation of constantMemory is subject to variability with the purpose to support variability in the software component or module implementations. Typically different algorithms in the implementation are requiring different number of memory objects. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
constantValueMapping	ConstantSpecificationMappingSet	*	ref	Reference to the ConstanSpecificationMapping to be applied for the particular InternalBehavior
dataTypeMapping	DataTypeMappingSet	*	ref	Reference to the DataTypeMapping to be applied for the particular InternalBehavior
exclusiveArea	ExclusiveArea	*	aggr	This specifies an ExclusiveArea for this InternalBehavior. The exclusiveArea is local to the component resp. module. The aggregation of ExclusiveAreas is subject to variability. Note: the number of ExclusiveAreas might vary due to the conditional existence of RunnableEntities or BswModuleEntities. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
exclusiveAreaNestingOrder	ExclusiveAreaNestingOrder	*	aggr	This represents the set of ExclusiveAreaNestingOrder owned by the InternalBehavior. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Attribute	Datatype	Mul.	Kind	Note
staticMemory	VariableDataPrototype	*	aggr	Describes a read and writeable static memory object representing measurment variables implemented by this software component. 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 DataElementPrototype 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 components implementations. Typically different algorithms in the implementation are requiring different number of memory objects. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table G.58: InternalBehavior

Class	≪atpVariation≫ **LinCommunicationController (abstract)**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Lin::LinTopology			
Note	LIN bus specific communication controller attributes.			
Base	ARObject,CommunicationController,Identifiable,MultilanguageReferrable,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
protocolVersion	String	1	attr	Version specifier for a communication protocol.

Table G.59: LinCommunicationController

Class	**MemorySection**			
Package	M2::AUTOSARTemplates::CommonStructure::ResourceConsumption::MemorySectionUsage			
Note	Provides a description of an abstract memory section used in the Implementation for code or data. It shall be declared by the Implementation Description of the module or component, which actually allocates the memory in its code. This means in case of data prototypes which are allocated by the RTE, that the generated Implementation Description of the RTE shall contain the corresponding MemorySections. The attribute "symbol" (if symbol is missing: "shortName") defines the module or component specific section name used in the code. For details see the document "Specification of Memory Mapping". Typically the section name is build according the pattern: <SwAddrMethod shortName>[_<further specialization nominator>][_<alignment>] where • **[<SwAddrMethod shortName>]** is the shortName of the referenced SwAddrMethod • **[_<further specialization nominator>]** is an optional infix to indicate the specialization in the case that several MemorySections for different purpose of the same Implementation Description referring to the same or equally named SwAddrMethods. • **[_<alignment>]** is the alignment attributes value and is only applicable in the case that the memoryAllocationKeywordPolicy value of the referenced SwAddrMethod is set to addrMethodShortNameAndAlignment 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 BswModuleDescription resp. the SwcComponentType. It can be superseded by the prefix attribute.			
Base	ARObject,Identifiable,MultilanguageReferrable,Referrable			
Attribute	***Datatype***	***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 ExecutableEntitities in different sections even if the associated SwAddrmethod is the same. This is applicable to code sections only.

Attribute	Datatype	Mul.	Kind	Note
memClass Symbol	CIdentifier	0..1	ref	Defines a specific symbol in order to generate the compiler abstraction "memclass" code for this MemorySection. 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	*	ref	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 BswModuleDescription'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.

Attribute	Datatype	Mul.	Kind	Note
swAddrmethod	SwAddrMethod	1	ref	This association indicates that this module specific (abstract) memory section is part of an overall SwAddrMethod, 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	ref	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 sectionNamePrefixes.

Table G.60: MemorySection

Class	**ModeDeclaration**			
Package	M2::AUTOSARTemplates::CommonStructure::ModeDeclaration			
Note	Declaration of one Mode. The name and semantics of a specific mode is not defined in the meta-model.			
Base	ARObject,AtpClassifier,AtpFeature,AtpStructureElement,Identifiable,MultilanguageReferrable,Referrable			
Attribute	**Datatype**	**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 ModeDeclaration.

Table G.61: ModeDeclaration

Class	**ModeDeclarationGroup**			
Package	M2::AUTOSARTemplates::CommonStructure::ModeDeclaration			
Note	A collection of Mode Declarations. Also, the initial mode is explicitly identified. **Tags:** atp.recommendedPackage=ModeDeclarationGroups			
Base	ARElement,ARObject,AtpBlueprint,AtpBlueprintable,AtpClassifier,AtpType,CollectableElement,Identifiable,MultilanguageReferrable,PackageableElement,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**

Attribute	Datatype	Mul.	Kind	Note
initialMode	ModeDeclaration	1	ref	The initial mode of the ModeDeclarationGroup. This mode is active before any mode switches occurred.
modeDeclaration	ModeDeclaration	1..*	aggr	The ModeDeclarations collected in this ModeDeclarationGroup. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=blueprintDerivationTime
modeManagerErrorBehavior	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
modeUserErrorBehavior	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).
onTransitionValue	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 G.62: ModeDeclarationGroup

Class	ModeDeclarationGroupPrototype			
Package	M2::AUTOSARTemplates::CommonStructure::ModeDeclaration			
Note	The ModeDeclarationGroupPrototype specifies a set of Modes (ModeDeclarationGroup) which is provided or required in the given context.			
Base	ARObject,AtpFeature,AtpPrototype,Identifiable,MultilanguageReferrable,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
swCalibrationAccess	SwCalibrationAccessEnum	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" (= ModeDeclarationGroup) supported by a component **Stereotypes:** isOfType

Table G.63: ModeDeclarationGroupPrototype

Class	NmPdu			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Network Management Pdu **Tags:** atp.recommendedPackage=Pdus			
Base	ARObject,CollectableElement,FibexElement,Identifiable,Multilanguage Referrable,PackageableElement,Pdu,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
iSignalToIPduMapping	ISignalToIPduMapping	*	aggr	This optional aggregation is used to describe NmUserData 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.
nmDataInformation	Boolean	0..1	attr	Defines if the Pdu contains NM Data. If the NmPdu does not aggregate any ISignalToIPduMappings it still may contain UserData that is set via Nm_SetUserData(). If the ISignalToIPduMapping exists then the nmDataInformation attribute shall be ignored.
nmVoteInformation	Boolean	0..1	attr	Defines if the Pdu contains NM Vote information.
unusedBitPattern	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 nmDataInformation attribute is set to true.

Table G.64: NmPdu

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	***Datatype***	***Mul.***	***Kind***	***Note***
value	Numerical	1	attr	This is the value itself. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table G.65: NumericalValueSpecification

Class	PPortPrototype			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components			
Note	Component port providing a certain port interface.			
Base	ARObject,AbstractProvidedPortPrototype,AtpBlueprintable,AtpFeature,Atp Prototype,Identifiable,MultilanguageReferrable,PortPrototype,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
providedInterface	PortInterface	1	tref	The interface that this port provides. **Stereotypes:** isOfType

Attribute	*Datatype*	*Mul.*	*Kind*	*Note*

Table G.66: PPortPrototype

Class	**PRPortPrototype**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components			
Note	This kind of PortPrototype can take the role of both a required and a provided PortPrototype.			
Base	ARObject,AbstractProvidedPortPrototype,AbstractRequiredPortPrototype,AtpBlueprintable,AtpFeature,AtpPrototype,Identifiable,MultilanguageReferrable,PortPrototype,Referrable			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
providedRequiredInterface	PortInterface	1	tref	This represents the PortInterface used to type the PRPortPrototype **Stereotypes:** isOfType

Table G.67: 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			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
–	–	–	–	–

Table G.68: PackageableElement

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	*Datatype*	*Mul.*	*Kind*	*Note*
constantMapping	ConstantSpecificationMappingSet	*	ref	Reference to the ConstanSpecificationMapping to be applied for the particular ParameterSwComponentType
dataTypeMapping	DataTypeMappingSet	*	ref	Reference to the DataTypeMapping to be applied for the particular ParameterSwComponentType

Attribute	Datatype	Mul.	Kind	Note
instantiationDataDefProps	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 G.69: 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	Datatype	Mul.	Kind	Note
providedOuterPort	AbstractProvidedPortPrototype	1	ref	This represents the provided outer delegation PortPrototype of the PassThroughSwConnector.
requiredOuterPort	AbstractRequiredPortPrototype	1	ref	This represents the required outer delegation PortPrototype of the PassThroughSwConnector.

Table G.70: 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			
Attribute	Datatype	Mul.	Kind	Note
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.
metaDataLength	PositiveInteger	0..1	attr	Number of additional bytes of MetaData in the PDU data field. The MetaData contains auxiliary information for the PDU, e.g. the CAN ID.

Table G.71: 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	***Datatype***	***Mul.***	***Kind***	***Note***
packingByteOrder	ByteOrderEnum	1	attr	This attribute defines the order of the bytes of the Pdu and the packing into the Frame. The byte ordering "Little Endian" (MostSignificantByteLast) and "Big Endian" (MostSignificantByteFirst) can be selected. A mix between Little Endian and Big Endian within a Frame is not allowed (all PduToFrameMappings within a Frame must have the same packingByteOrder).
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 packingByteOrder 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 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.
updateIndicationBitPosition	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 updateIndicationBitPosition is linked to the value of the attribute packingByteOrder because the method of finding the bit position is different for the values mostSignificantByteFirst and mostSignificantByteLast. This means that if the value of packingByteOrder is changed while the value of updateIndicationBitPosition remains unchanged the exact bit position of updateIndicationBitPosition within the enclosing Frame still undergoes a change.

Table G.72: PduToFrameMapping

Class	**PhysConstrs**			
Package	M2::AUTOSARTemplates::CommonStructure::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	***Datatype***	***Mul.***	***Kind***	***Note***
lowerLimit	Limit	0..1	ref	This specifies the lower limit of the constraint. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=20
maxDiff	Numerical	0..1	attr	Maximum difference that is permitted between two consecutive values if the constraint is applied to an axis. **Tags:** xml.sequenceOffset=60
maxGradient	Numerical	0..1	attr	This element specifies the maximum slope that may be used in curves and maps. **Tags:** xml.sequenceOffset=50
monotony	MonotonyEnum	0..1	attr	This specifies the monotony constraints on the data object. Note that this applies only to curves and maps. **Tags:** xml.sequenceOffset=70
scaleConstr (ordered)	ScaleConstr	*	aggr	This is one particular scale which contributes to the data constraints. **Tags:** xml.roleElement=true; xml.roleWrapperElement=true; xml.sequenceOffset=40; xml.typeElement=false; xml.typeWrapperElement=false
unit	Unit	0..1	ref	This is the unit to which the physical constraints relate to. In particular, it is the physical unit of the specified limits. **Tags:** xml.sequenceOffset=80
upperLimit	Limit	0..1	ref	This specifies the upper limit of the constraint. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=30

Table G.73: PhysConstrs

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	Datatype	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 port of this component 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 G.74: 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			
Attribute	Datatype	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 or • 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 G.75: PortInterface

Class	**PortInterfaceMapping (abstract)**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	Specifies one PortInterfaceMapping to support the connection of Ports typed by two different PortInterfaces with PortInterface elements having unequal names and/or unequal semantic (resolution or range).			
Base	ARObject,AtpBlueprint,AtpBlueprintable,Identifiable,Multilanguage Referrable,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
–	–	–	–	–

Table G.76: 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,Multilanguage Referrable,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
clientServerAnnotation	ClientServerAnnotation	*	aggr	Annotation of this PortPrototype with respect to client/server communication.
delegatedPortAnnotation	DelegatedPortAnnotation	0..1	aggr	Annotations on this delegated port.
ioHwAbstractionServerAnnotation	IoHwAbstractionServerAnnotation	*	aggr	Annotations on this IO Hardware Abstraction port.
modePortAnnotation	ModePortAnnotation	*	aggr	Annotations on this mode port.
nvDataPortAnnotation	NvDataPortAnnotation	*	aggr	Annotations on this non voilatile data port.
parameterPortAnnotation	ParameterPortAnnotation	*	aggr	Annotations on this parameter port.
senderReceiverAnnotation	SenderReceiverAnnotation	*	aggr	Collection of annotations of this ports sender/receiver communication.
triggerPortAnnotation	TriggerPortAnnotation	*	aggr	Annotations on this trigger port.

Table G.77: PortPrototype

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	Datatype	Mul.	Kind	Note
requiredInterface	PortInterface	1	tref	The interface that this port requires, i.e. the port depends on another port providing the specified interface. **Stereotypes:** isOfType

Table G.78: 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			
Attribute	Datatype	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 G.79: RTEEvent

Class	ReceiverComSpec (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	Receiver-specific communication attributes (RPortPrototype typed by SenderReceiverInterface).			
Base	ARObject,RPortComSpec			
Attribute	Datatype	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 ApplicationCompositeDataTypes.
dataElement	VariableDataPrototype	1	ref	Data element these attributes belong to.
externalReplacement	AutosarDataPrototype	0..1	ref	This reference is used to reference the AutosarDataPrototype to be taken for sourcing an external replacement in the out-of-range handling.

Attribute	Datatype	Mul.	Kind	Note
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 MaxDeltaCounterInit 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 dataElement is mapped to a communication bus.
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 behaviour of a received counter.
usesEndToEndProtection	Boolean	1	attr	This indicates whether the corresponding dataElement shall be transmitted using end-to-end protection. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table G.80: ReceiverComSpec

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			
Attribute	Datatype	Mul.	Kind	Note

Attribute	Datatype	Mul.	Kind	Note
shortName	Identifier	1	ref	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

Table G.81: Referrable

Class	**RoleBasedPortAssignment**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::ServiceMapping			
Note	This class specifies an assignment of a role to a particular service port (RPortPrototype or PPortPrototype) 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	**Datatype**	**Mul.**	**Kind**	**Note**
portPrototype	PortPrototype	1	ref	Service port used in the assigned role. This port shall either belong to the same AtomicSoftwareComponent as the SwcInternalBehavior which owns the ServiceDependency or to the same NvBlockComponentType as the NvBlockDescriptor.
role	Identifier	1	ref	This is the role of the assigned Port in the given context. The value shall be a shortName of the Blueprint of a PortInterface as standardized in the Software Specification of the related AUTOSAR Service.

Table G.82: RoleBasedPortAssignment

Class	**RunnableEntity**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior			
Note	A RunnableEntity represents the smallest code-fragment that is provided by an AtomicSwComponentType 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,MultilanguageReferrable,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
argument (ordered)	RunnableEntityArgument	*	aggr	This represents the formal definition of a an argument to a RunnableEntity.

Attribute	Datatype	Mul.	Kind	Note
asynchronousServerCallResultPoint	AsynchronousServerCallResultPoint	*	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:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
canBeInvokedConcurrently	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 AtomicSwComponentType). 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".
dataReadAccess	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:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
dataReceivePointByArgument	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:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Attribute	Datatype	Mul.	Kind	Note
dataReceivePointByValue	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:** atpVariation **Tags:** 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:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
dataWriteAccess	VariableAccess	*	aggr	RunnableEntity has implicit write access to dataElement of a sender-receiver PortPrototype or nv data of a nv data PortPrototype. The aggregation of dataWriteAccess is subject to variability with the purpose to support the conditional existence of sender receiver ports or the variant existence of dataWriteAccess in the implementation. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
externalTriggeringPoint	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:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
internalTriggeringPoint	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:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Attribute	Datatype	Mul.	Kind	Note
modeAccessPoint	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:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
modeSwitchPoint	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:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
parameterAccess	ParameterAccess	*	aggr	The presence of a ParameterAccess implies that a RunnableEntity needs read only access to a ParameterDataPrototype which may either be local or within a PortPrototype. 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 ParameterAccess (points) in the implementation. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
readLocalVariable	VariableAccess	*	aggr	The presence of a readLocalVariable implies that a RunnableEntity needs read access to a VariableDataPrototype 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 explicitInterRunnableVariable or the variant existence of readLocalVariable (points) in the implementation. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Attribute	Datatype	Mul.	Kind	Note
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:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
symbol	CIdentifier	1	ref	The symbol describing this RunnableEntity's entry point. This is considered the API of the RunnableEntity and is required during the RTE contract phase.
waitPoint	WaitPoint	*	aggr	The WaitPoint associated with the RunnableEntity.
writtenLocalVariable	VariableAccess	*	aggr	The presence of a writtenLocalVariable implies that a RunnableEntity needs write access to a VariableDataPrototype 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 explicitInterRunnableVariable or the variant existence of writtenLocalVariable (points) in the implementation. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table G.83: RunnableEntity

Class	**SenderComSpec (abstract)**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Communication			
Note	Communication attributes for a sender port (PPortPrototype typed by SenderReceiverInterface).			
Base	ARObject,PPortComSpec			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
compositeNetworkRepresentation	CompositeNetworkRepresentation	*	aggr	This represents a CompositeNetworkRepresentation defined in the context of a SenderComSpec.
dataElement	VariableDataPrototype	1	ref	Data element these quality of service attributes apply to.
handleOutOfRange	HandleOutOfRangeEnum	1	attr	This attribute controls how out-of-range values shall be dealt with.
networkRepresentation	SwDataDefProps	0..1	aggr	A networkRepresentation is used to define how the dataElement is mapped to a communication bus.

Attribute	Datatype	Mul.	Kind	Note
transmissionAcknowledge	TransmissionAcknowledgementRequest	0..1	aggr	Requested transmission acknowledgement for data element.
usesEndToEndProtection	Boolean	1	attr	This indicates whether the corresponding dataElement shall be transmitted using end-to-end protection. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table G.84: SenderComSpec

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	***Datatype***	***Mul.***	***Kind***	***Note***
dataElement	VariableDataPrototype	1..*	aggr	The data elements of this SenderReceiverInterface.
invalidationPolicy	InvalidationPolicy	*	aggr	InvalidationPolicy for a particular dataElement

Table G.85: SenderReceiverInterface

Class	**SensorActuatorSwComponentType**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components			
Note	The SensorActuatorSwComponentType introduces the possibility to link from the software representation of a sensor/actuator to its hardware description provided by the ECU Resource Template. **Tags:** atp.recommendedPackage=SwComponentTypes			
Base	ARElement,ARObject,AtomicSwComponentType,AtpBlueprint,AtpBlueprintable,AtpClassifier,AtpType,CollectableElement,Identifiable,MultilanguageReferrable,PackageableElement,Referrable,SwComponentType			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
sensorActuator	HwDescriptionEntity	1	ref	Reference from the Sensor Actuator Software Component Type to the description of the actual hardware.

Table G.86: SensorActuatorSwComponentType

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			
Attribute	Datatype	Mul.	Kind	Note
–	–	–	–	–

Table G.87: 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,AtpType,CollectableElement,Identifiable,MultilanguageReferrable,PackageableElement,Referrable,SwComponentType			
Attribute	Datatype	Mul.	Kind	Note
–	–	–	–	–

Table G.88: ServiceProxySwComponentType

Class	**SignalPathConstraint (abstract)**			
Package	M2::AUTOSARTemplates::SystemTemplate::SignalPaths			
Note	Additional guidelines for the System Generator, which specific way a signal between two Software Components should take in the network without defining in which frame and with which timing it is transmitted.			
Base	ARObject			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
introduction	DocumentationBlock	1	aggr	This represents introductory documentation about the signal path constraint.

Table G.89: SignalPathConstraint

Class	**StackUsage (abstract)**			
Package	M2::AUTOSARTemplates::CommonStructure::ResourceConsumption::StackUsage			
Note	Describes the stack memory usage of a software.			
Base	ARObject,Identifiable,MultilanguageReferrable,Referrable			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
executableEntity	ExecutableEntity	0..1	ref	The executable entity for which this stack usage is described.
hardwareConfiguration	HardwareConfiguration	0..1	aggr	Contains information about the hardware context this stack usage is describing.
hwElement	HwElement	0..1	ref	Specifies for which hardware element (e.g. ECU) this stack usage is given.
softwareContext	SoftwareContext	0..1	aggr	Contains details about the software context this stack usage is provided for.

Table G.90: StackUsage

Class	**SwBaseType**			
Package	M2::AUTOSARTemplates::CommonStructure::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	*Datatype*	*Mul.*	*Kind*	*Note*
–	–	–	–	–

Table G.91: SwBaseType

Class	**SwComponentPrototype**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Composition			
Note	Role of a software component within a composition.			
Base	ARObject,AtpFeature,AtpPrototype,Identifiable,MultilanguageReferrable,Referrable			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*

Attribute	Datatype	Mul.	Kind	Note
type	SwComponentType	1	tref	Type of the instance. **Stereotypes:** isOfType

Table G.92: 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			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
consistencyNeeds	ConsistencyNeeds	*	aggr	This represents the colelction of ConsistencyNeeds owned by the enclosing SwComponentType. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
port	PortPrototype	*	aggr	The ports through which this component 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
swComponentDocumentation	SwComponentDocumentation	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 G.93: 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,Multilanguage Referrable,Referrable			
Attribute	**Datatype**	**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 G.94: SwConnector

Class	≪atpVariation≫ SwDataDefProps			
Package	M2::AUTOSARTemplates::CommonStructure::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 DataTypes in the programming language (or in AUTOSAR). This is mainly expressed by properties like swRecordLayout and swCalprmAxisSet • Implementation aspects, mainly expressed by swImplPolicy, swVariableAccessImplPolicy, swAddrMethod, swPointerTagetProps, baseType, implementationDataType and additionalNativeTypeQualifier • 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			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**

Attribute	Datatype	Mul.	Kind	Note
additionalNativeTypeQualifier	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

Attribute	Datatype	Mul.	Kind	Note
implementationDataType	ImplementationDataType	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 SwPointerTargetProps), 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
mcFunction	Identifier	0..1	ref	Specifies the name of a "Function" (in the sense of the MC system) to which this data object belongs. This corresponds to the Function in ASAM MCD 2MC /ASAP2 which defines the characteristic resp. which provides the measurement as output. The function name is only used for support of MC systems. It can be predefined on the level of software component design. If it is not predefined, it could be filled out with a reasonable name, e.g. the component prototype name, from the ECU extract. Note: This attribute is deprecated because an explicit model of MC functions can be set up by using the meta-class McFunction. **Tags:** atp.Status=obsolete xml.sequenceOffset=257
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

Attribute	Datatype	Mul.	Kind	Note
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 memoryAllocationKeywordPolicy of the referenced SwAddrMethod. **Tags:** xml.sequenceOffset=33
swBitRepresentation	SwBitRepresentation	0..1	aggr	Description of the binary representaion in case of a bit variable. **Tags:** xml.sequenceOffset=60
swCalibrationAccess	SwCalibrationAccessEnum	0..1	attr	Specifies the read or write access by MCD tools for this data object. **Tags:** xml.sequenceOffset=70
swCalprmAxisSet	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
swComparisonVariable	SwVariableRefProxy	*	aggr	Variables used for comparison in an MCD process. **Tags:** xml.sequenceOffset=170; xml.typeElement=false
swDataDependency	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

Attribute	Datatype	Mul.	Kind	Note
swIntendedResolution	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
swInterpolationMethod	Identifier	0..1	ref	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
swPointerTargetProps	SwPointerTargetProps	0..1	aggr	Specifies that the containing data object is a pointer to another data object. **Tags:** xml.sequenceOffset=280
swRecordLayout	SwRecordLayout	0..1	ref	Record layout for this data object. **Tags:** xml.sequenceOffset=290
swRefreshTiming	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

Attribute	Datatype	Mul.	Kind	Note
swTextProps	SwTextProps	0..1	aggr	the specific properties if the data object is a text object. **Tags:** xml.sequenceOffset=120
swValueBlockSize	Numerical	0..1	attr	This represents the size of a Value Block **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=80
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
valueAxisDataType	ApplicationPrimitiveDataType	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 G.95: SwDataDefProps

Enumeration	**SwImplPolicyEnum**
Package	M2::AUTOSARTemplates::CommonStructure::DataDefProperties
Note	Specifies the implementation strategy with respect to consistency mechanisms of variables.
Literal	***Description***
const	forced implementation such that the running software within the ECU shall not modify it. For example implemented with the "const" modifier in C. This can be applied for parameters (not for those in NvRam) as well as argument data prototypes.
fixed	This data element is fixed. In particular this indicates, that it might also be implemented e.g. as in place data, (#DEFINE).
measurementPoint	The data element is created for measurement purposes only. The data element is never read directly within the ECU software. In contrast to a "standard" data element in an unconnected provide port is, this unconnection is guaranteed for measurementPoint data elements.
queued	The content of the data element is queued and the data element has 'event' semantics, i.e. data elements are stored in a queue and all data elements are processed in 'first in first out' order. The queuing is intended to be implemented by RTE Generator. This value is not applicable for parameters.
standard	This is applicable for all kinds of data elements. For variable data prototypes the 'last is best' semantics applies. For parameter there is no specific implementation directive.

Table G.96: SwImplPolicyEnum

Class	SwRecordLayout			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::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,Multilanguage Referrable,PackageableElement,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
swRecordLayoutGroup	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 G.97: SwRecordLayout

Class	SwRecordLayoutGroup			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::RecordLayout			
Note	Specifies how a record layout is set up. Using SwRecordLayoutGroup it recursively models iterations through axis values. The subelement swRecordLayoutGroupContentType may reference other SwRecordLayouts, SwRecordLayoutVs and SwRecordLayoutGroups for the modeled record layout.			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
desc	MultiLanguageOverviewParagraph	0..1	aggr	This property allows a brief description about the particular record layout group which can help to identify the entry. In-depth documentation should go to introduction of the surrounding record layout. **Tags:** xml.sequenceOffset=20
category	AsamRecordLayoutSemantics	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 recordlayout keywords in swRecordlayoutGroup but not always vice versa. Therefore the mapping is provided in this optional attribute. **Tags:** xml.sequenceOffset=5
shortLabel	Identifier	1	ref	This attribute specifies a name which can be used e.g. when ECU code is generated from the record layout group. **Tags:** xml.sequenceOffset=3

Attribute	Datatype	Mul.	Kind	Note
swGenericAxisParamType	SwGenericAxisParamType	0..1	ref	This association allows to specify record layout groups to iterate over generic axis parameters. For example, if the generic axis parameter is an array, the record layout group will iterate over this array. Obviously, the axis referred to by swRecordLayoutGroupAxis shall be a generic axis in which the referenced SwGenericAxisType is aggregated. **Tags:** xml.sequenceOffset=50
swRecordLayoutComponent	Identifier	0..1	ref	is used to denote the component to which the group in question applies. Thus, the record layout supports structured objects.This secures independence from the sequence of components, because they can be referred to via name. **Tags:** xml.sequenceOffset=90
swRecordLayoutGroupAxis	AxisIndexType	0..1	attr	This attribute specifies the iteration axis number for a SwRecordLayoutGroup. The current record layout group then refers exactly to the axis with this number. This means that the values are taken by iterating along the thus referenced axis. **Tags:** xml.sequenceOffset=30
swRecordLayoutGroupContentType	SwRecordLayoutGroupContent	0..1	aggr	This is the contents of the recordLayout which is produced for every step of iteration. **Tags:** xml.roleElement=false; xml.roleWrapperElement=false; xml.sequenceOffset=100; xml.typeElement=false; xml.typeWrapperElement=false
swRecordLayoutGroupFrom	RecordLayoutIteratorPoint	0..1	attr	This element specifies the iterator index for the point in the axis from which a record layout group is commenced. Negative values are also possible, i.e. the value -4 counts from the fourth value from the end. If this property is missing, the iteration starts with '1'. **Tags:** xml.sequenceOffset=60
swRecordLayoutGroupIndex	NameToken	0..1	attr	This element attributes a symbolic name to the iterator of the superimposed record layout group. This can be referenced as a loop index in contained SwRecordLayoutV elements. **Tags:** xml.sequenceOffset=40
swRecordLayoutGroupStep	Integer	0..1	attr	This property specifies the step width for the iterator index that is used for the current record layout group. Note that negative values are also possible, in case of the starting point is higher than the endpoint. If the property is missing, the step width is "1". **Tags:** xml.sequenceOffset=80

Attribute	Datatype	Mul.	Kind	Note
swRecordLayoutGroupTo	RecordLayoutIteratorPoint	0..1	attr	This element specifies the end point for the iteration. Negative values are also possible, i.e. the value -4 counts up to the fourth value from the end. If this property is not there, the iteration ends at "-1" which is the last element. Note that depending on the arraySizeSemantics of SwTextProps the iteration ends at the value specified in swMaxTextSize. **Tags:** xml.sequenceOffset=70

Table G.98: SwRecordLayoutGroup

Class	SwSystemconst			
Package	M2::AUTOSARTemplates::CommonStructure::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	**Datatype**	**Mul.**	**Kind**	**Note**
swDataDefProps	SwDataDefProps	0..1	aggr	This denotes the data defintion properties of the system constant. In particular it is the limits and - in case the system constant is an enumeration - the compu method. **Tags:** xml.sequenceOffset=40

Table G.99: SwSystemconst

Class	SwSystemconstValue			
Package	M2::AUTOSARTemplates::GenericStructure::VariantHandling			
Note	This meta-class assigns a particular value to a system constant.			
Base	ARObject			
Attribute	**Datatype**	**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

Attribute	Datatype	Mul.	Kind	Note
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 SwSystemconst as it is processed in the Formula Language. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=20

Table G.100: 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,Multilanguage Referrable,PackageableElement,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
swSystemconstantValue	SwSystemconstValue	*	aggr	This is one particular value of a system constant.

Table G.101: SwSystemconstantValueSet

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,Multilanguage Referrable,PackageableElement,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
behavior	SwcInternalBehavior	1	ref	The internal behavior implemented by this Implementation.
perInstanceMemorySize	PerInstanceMemorySize	*	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 PerInstanceMemory. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Attribute	Datatype	Mul.	Kind	Note
requiredRTEVendor	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 G.102: 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,MultilanguageReferrable,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
arTypedPerInstanceMemory	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 components implementations. Typically different algorithms in the implementation are requiring different number of memory objects. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
event	RTEEvent	*	aggr	This is a RTEEvent specified for the particular SwcInternalBehavior. 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 DataReceivedEvents or due to different scheduling needs of algorithms. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Attribute	Datatype	Mul.	Kind	Note
explicitInterRunnableVariable	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:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
handleTerminationAndRestart	HandleTerminationAndRestartEnum	1	attr	This attribute controls the behavior with respect to stopping and restarting. The corresponding AtomicSwComponentType may either not support stop and restart, or support only stop, or support both stop and restart.
implicitInterRunnableVariable	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:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
includedDataTypeSet	IncludedDataTypeSet	*	aggr	The includedDataTypeSet is used by a software component for its implementation.
includedModeDeclarationGroupSet	IncludedModeDeclarationGroupSet	*	aggr	This aggregation represents the included ModeDeclarationGroups
instantiationDataDefProps	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 and component local memories like "perInstanceParameter" or "arTypedPerInstanceMemory". **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Attribute	Datatype	Mul.	Kind	Note
perInstanceMemory	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:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
perInstanceParameter	ParameterDataPrototype	*	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:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
runnable	RunnableEntity	1..*	aggr	This is a RunnableEntity specified for the particular SwcInternalBehavior. 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 PortPrototypes using DataReceivedEvents or due to different scheduling needs of algorithms. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Attribute	Datatype	Mul.	Kind	Note
serviceDependency	SwcServiceDependency	*	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 SwcInternalBehavior can be located in a different physical file in order to support that SwcServiceDependency might be provided in later development steps or even by different expert domain (e.g OBD expert for Obd related Service Needs) tools. Therefore the aggregation is «atpSplitable». **Stereotypes:** atpVariation **Tags:** atp.Splitkey=serviceDependency.shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
sharedParameter	ParameterDataPrototype	*	aggr	Defines parameter(s) or characteristic value(s) shared between SwComponentPrototypes of the same SwComponentType 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
supportsMultipleInstantiation	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).
variationPointProxy	VariationPointProxy	*	aggr	Proxy of a variation points in the C/C++ implementation.

Table G.103: SwcInternalBehavior

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	***Datatype***	***Mul.***	***Kind***	***Note***

Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
value	VerbatimString	1	ref	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 G.104: TextValueSpecification

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,Multilanguage Referrable,Referrable			
Attribute	*Datatype*	*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 G.105: 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	*Datatype*	*Mul.*	*Kind*	*Note*
trigger	Trigger	1..*	aggr	The Trigger of this trigger interface.

Table G.106: TriggerInterface

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	**Datatype**	**Mul.**	**Kind**	**Note**
initValue	ValueSpecification	0..1	aggr	Specifies initial value(s) of the VariableDataPrototype

Table G.107: 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 postBuildVariantCriterion is fulfilled.			
Base	ARObject			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
desc	MultiLanguageOverviewParagraph	0..1	aggr	This allows to describe shortly the purpose of the variation point. **Tags:** xml.sequenceOffset=20
blueprintCondition	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
formalBlueprintCondition	BlueprintFormula	0..1	aggr	This denotes a formal blueprintCondition. This shall be not in contradiction with blueprintCondition. It is recommanded only to use one of the two. **Tags:** xml.sequenceOffset=29
postBuildVariantCondition	PostBuildVariantCondition	*	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

Attribute	Datatype	Mul.	Kind	Note
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	ref	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 VariationPoint. Note that the mulitplicity is 0..1 in order to support pure postBuild variants. **Tags:** xml.sequenceOffset=30

Table G.108: VariationPoint

Class	ViewMap			
Package	M2::AUTOSARTemplates::GenericStructure::ViewMapSet			
Note	The ViewMap allows to relate any number of elements on the "first" side to any number of elements on the "second" side. Since the ViewMap does not address a specific mapping use-case the roles "first" and "second" shall imply this generality. This mapping allows to trace transformations of artifacts within the AUTOSAR environment. The references to the mapped elements can be plain references and/or InstanceRefs.			
Base	ARObject,Identifiable,MultilanguageReferrable,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
firstElement	Referrable	*	ref	Reference to identifible elements on the first "side". **Tags:** xml.sequenceOffset=20
firstElementInstance	AtpFeature	*	iref	InstanceRefs to elements on the first "side". **Tags:** xml.sequenceOffset=50
role	Identifier	0..1	ref	This attribute is used to describe specific mapping scenarios, e.g. the mappings: AR_AbstractSystemDescription_SystemDescription AR_SystemDescription_SystemExtract **Tags:** xml.sequenceOffset=10

Attribute	Datatype	Mul.	Kind	Note
secondElement	Referrable	*	ref	Reference to identifible elements on the second "side". **Tags:** xml.sequenceOffset=30
secondElementInstance	AtpFeature	*	iref	InstanceRefs to elements on the second "side". **Tags:** xml.sequenceOffset=60

Table G.109: ViewMap

Class	**ViewMapSet**			
Package	M2::AUTOSARTemplates::GenericStructure::ViewMapSet			
Note	Collection of ViewMaps that are used to establish relationships between different AUTOSAR artifacts. **Tags:** atp.recommendedPackage=ViewMapSets			
Base	ARElement,ARObject,CollectableElement,Identifiable,MultilanguageReferrable,PackageableElement,Referrable			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
viewMap	ViewMap	*	aggr	ViewMaps that are collected by the ViewMapSet.

Table G.110: ViewMapSet