Document Title	Specification of COM Based Transformer
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	662
Document Classification	Standard

Document Status	Final
Part of AUTOSAR Release	4.2.2

Document Change History		
Release	**Changed by**	**Description**
4.2.2	AUTOSAR Release Management	• Exclude support for external trigger communication [SWS_ComXf_00032]
4.2.1	AUTOSAR Release Management	• 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

1 Introduction and functional overview

The transformer feature provides functionality to chain arbitrary transformers when sending and receiving data in the RTE. The COM Based Transformer provides this functionality when the target bus system uses a fixed communication matrix with packed data representations.

2 Acronyms and Abbreviations

No specific terms have been introduced additionally to those already defined in [1].

3 Related documentation

3.1 Input documents

Bibliography

[1] Glossary
AUTOSAR_TR_Glossary

[2] General Specification on Transformers
AUTOSAR_ASWS_TransformerGeneral

[3] Specification of RTE Software
AUTOSAR_SWS_RTE

[4] Specification of Communication
AUTOSAR_SWS_COM

[5] General Requirements on Basic Software Modules
AUTOSAR_SRS_BSWGeneral

[6] Requirements on AUTOSAR Features
AUTOSAR_RS_Features

[7] System Template
AUTOSAR_TPS_SystemTemplate

3.2 Related standards and norms

Not applicable.

3.3 Related specification

AUTOSAR provides a General Specification on Transformers [2], which is also valid for COM Based Transformer.

Thus, the specification ASWS Transformer General shall be considered as additional and required specification for COM Based Transformer.

4 Constraints and assumptions

4.1 Limitations

For the COM Based Transformer all general transformer limitations (see [2]) apply.

Additionally the following restrictions apply for the COM Based Transformer:

[SWS_ComXf_00017] ⌈ The COM Based Transformer does not support Client-Server communication. ⌋*(SRS_Xfrm_00201)*

[SWS_ComXf_00032] ⌈ The COM Based Transformer does not support external trigger communication. ⌋*(SRS_Xfrm_00201)*

[SWS_ComXf_00018] ⌈ The COM Based Transformer only supports composite data types (i.e. Signal Groups from COM). ⌋*(SRS_Xfrm_00201)*

[SWS_ComXf_00019] ⌈ The COM Based Transformer only supports fix sized data types. ⌋*(SRS_Xfrm_00201)*

[SWS_ComXf_00022] ⌈ The COM Based Transformer only supports signal groups which are byte aligned. ⌋*(SRS_Xfrm_00201)*

[SWS_ComXf_00023] ⌈ The COM Based Transformer only supports signal groups where all group signals are mapped consecutively to the IPdu. ⌋*(SRS_Xfrm_00201)*

4.2 Applicability to car domains

The COM Based Transformer can be used for all domain applications when a fixed communication matrix is used.

5 Dependencies to other modules

The AUTOSAR RTE [3] has to exist to execute the COM Based Transformer.

The AUTOSAR COM configuration [4] of the data handled by the COM Based Transformer has to exists in order to allow the configuration of the COM Based Transformer.

5.1 File structure

5.1.1 Code file structure

The source code file structure is defined in the [2].

5.1.2 Header file structure

The header file structure of the COM Based Transformer is shown in Figure 5.1.

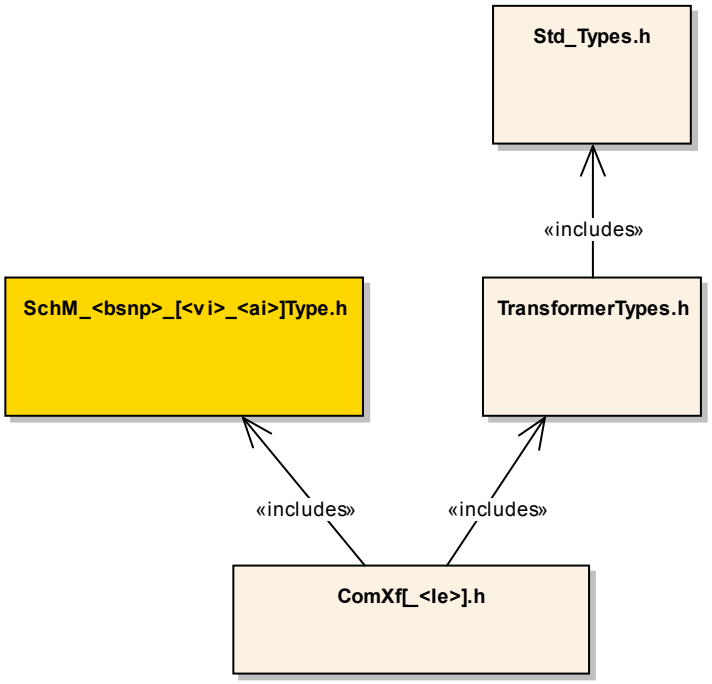

Figure 5.1: Header File Structure of COM Based Transformer

[SWS_ComXf_00001] ⌈ The header file `ComXf[_<Ie>].h` shall be the main include file for the COM Based transformer and include `TransformerTypes.h` and its Module Interlink Types Header file `SchM_<bsnp>_[<vi>_<ai>]Type.h` where
`<Ie>` is the optional implementation specific file name extension according [SWS_BSW_00103],
`<bsnp>` is the BSW Scheduler Name Prefix according [SWS_Rte_07593] and

[SWS_Rte_07594],
<vi> is the vendorId of the BSW module and
<ai> is the vendorApiInfix of the BSW module. ⌋(*SRS_BSW_00346*)

The file TransformerTypes.h contains the general transformer data types.

6 Requirements Tracing

The following table references the features specified in [5] and [6] and links to the fulfillments of these.

Feature	Description	Satisfied by
[SRS_BSW_00159]	All modules of the AUTOSAR Basic Software shall support a tool based configuration	[SWS_ComXf_00025]
[SRS_BSW_00337]	Classification of development errors	[SWS_ComXf_00028]
[SRS_BSW_00346]	All AUTOSAR Basic Software Modules shall provide at least a basic set of module files	[SWS_ComXf_00001]
[SRS_BSW_00404]	BSW Modules shall support post-build configuration	[SWS_ComXf_00030]
[SRS_BSW_00407]	Each BSW module shall provide a function to read out the version information of a dedicated module implementation	[SWS_ComXf_00024] [SWS_ComXf_00026] [SWS_ComXf_00027]
[SRS_BSW_00411]	All AUTOSAR Basic Software Modules shall apply a naming rule for enabling/disabling the existence of the API	[SWS_ComXf_00024] [SWS_ComXf_00026] [SWS_ComXf_00027]
[SRS_BSW_00441]	Naming convention for type, macro and function	[SWS_ComXf_00030]
[SRS_Xfrm_00009]	A fixed set of transformer classes shall exist	[SWS_ComXf_00003]
[SRS_Xfrm_00011]	A transformer shall belong to a specific transformer class	[SWS_ComXf_00003] [SWS_ComXf_00006]
[SRS_Xfrm_00201]	The COM Based Transformer shall define the serialization of atomic and structured data elements into linear arrays based on a fixed data mapping	[SWS_ComXf_00004] [SWS_ComXf_00005] [SWS_ComXf_00007] [SWS_ComXf_00008] [SWS_ComXf_00009] [SWS_ComXf_00010] [SWS_ComXf_00011] [SWS_ComXf_00012] [SWS_ComXf_00013] [SWS_ComXf_00014] [SWS_ComXf_00015] [SWS_ComXf_00016] [SWS_ComXf_00017] [SWS_ComXf_00018] [SWS_ComXf_00019] [SWS_ComXf_00020] [SWS_ComXf_00021] [SWS_ComXf_00022] [SWS_ComXf_00023] [SWS_ComXf_00032]
[SRS_Xfrm_00202]	The COM Based Transformer shall take its configuration from the COM module	[SWS_ComXf_00005] [SWS_ComXf_00014] [SWS_ComXf_00020] [SWS_ComXf_00025] [SWS_ComXf_00031]

7 Functional specification

When a SWC initiates an inter-ECU communication which is configured to be transformed, the SWC hands the data over to the RTE. The RTE executes the configured transformer chain which contains - if the configuration demands this - the COM Based Transformer.

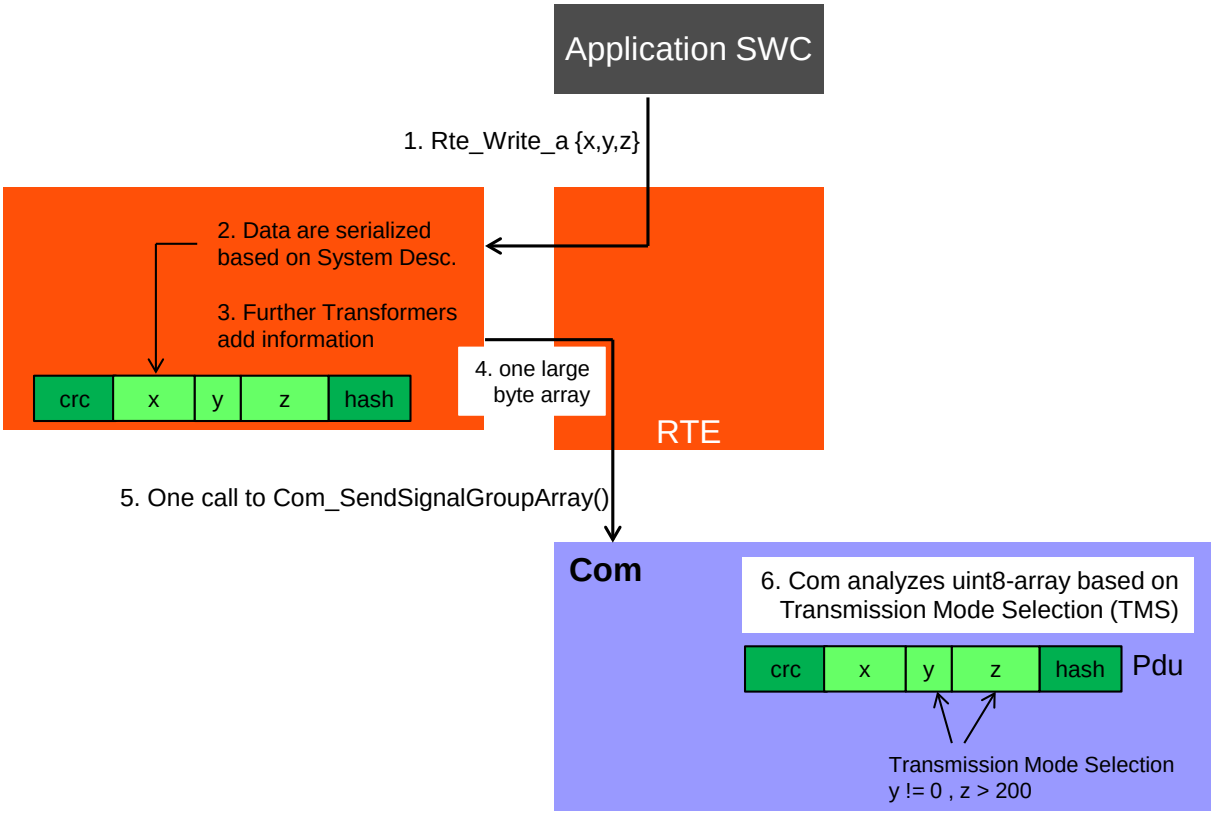

Figure 7.1: Overview of COM Based Transformer

The COM Based Transformer on the sender side serializes the data of the SWC and brings them into an uint8-array representation based on the communication matrix description. The uint8-array representation is forwarded to the COM module to be placed inside the respective IPdu. The COM module may analyze (depending on the configuration of the Transmission Mode Selection – TMS) the provided uint8-array and trigger the respective transmission mode. The IPdu is sent via the communication stack over the bus to the receiver(s).

The RTE of the receiver side executes the transformer chain in the reverse order. The COM Based transformer of the receiver deserializes the linear data back into the original data structure. These are handed over to the receiving SWC.

From the SWC's point of view it is totally transparent whether data are transformed or not.

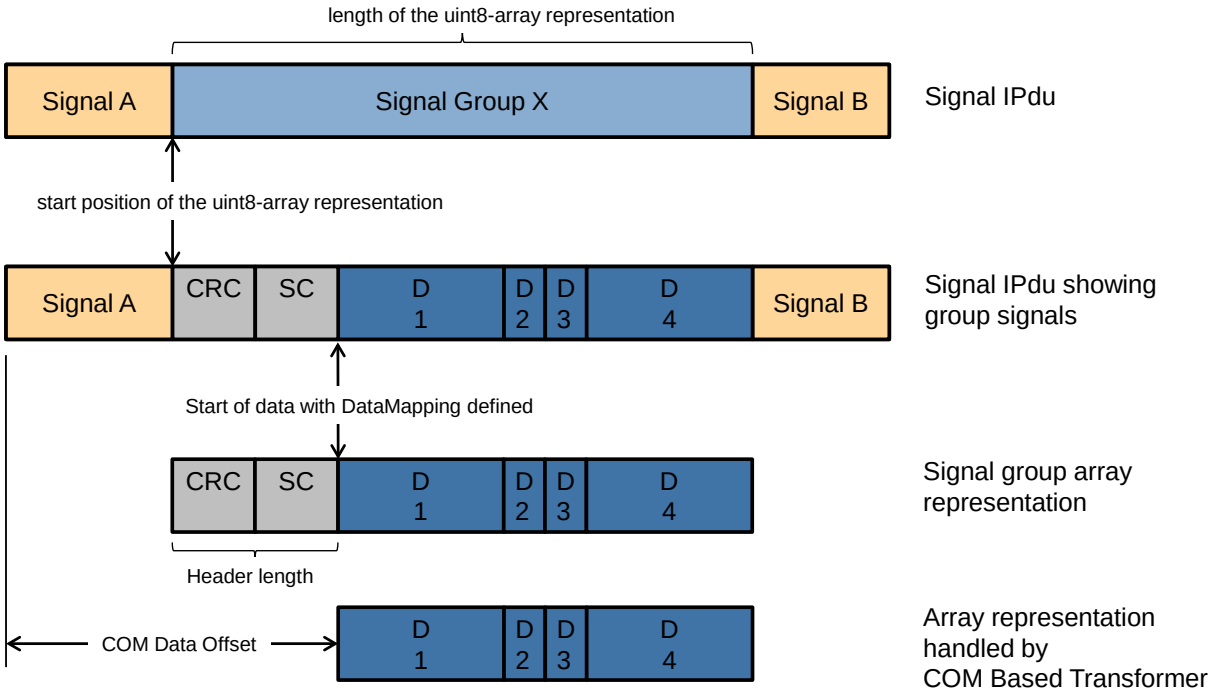

Figure 7.2: IPdu and signal layout

The handling of the data inside COM's IPdu buffer and the transformer buffer is shown in figure 7.2.

The `ISignalIPdu` is handled by the COM module as `ComIPdu` and may contain several parts (signals and signal groups). A signal group in COM is represented by the `ComSignalGroup` container.

In the System Template [7] it is possible to define that a signal group shall be handled by the COM Based Transformer. The usage of the COM Based Transformer for a specific transformer chain is defined by the reference `comBasedSignalGroupTransformation` from the `ISignalGroup` to `DataTransformation` (see figure 7.3).

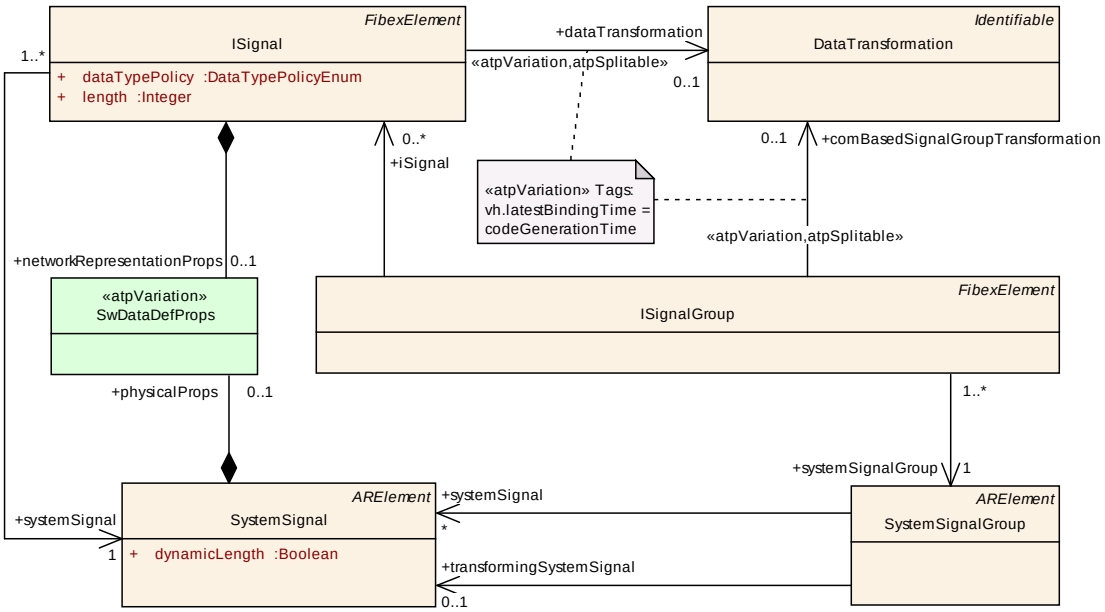

Figure 7.3: System Template Transformed communication

In the Ecu configuration of the COM module such `ComSignalGroup`s have a `ComSignalGroupArrayAccess=true` parameter defined.

If the `ComSignalGroupArrayAccess=true` then it implicitly defines the length of the signal group ([SWS_Com_00845] [4]) and the start position inside the `ComIPdu` where the signal group starts ([SWS_Com_00844] [4]). Thus there can be several signals and signal groups defined inside an `ISignalIPdu` (e.g. 'Signal A' and 'Signal B' are part of the IPdu but are not considered by the COM Based Transformer for the 'Signal Group X').

The Com APIs `Com_SendSignalGroupArray` and `Com_ReceiveSignalGroupArray` handle the signal group as array representation based on length of the signal group and the start position inside the `ComIPdu`. This array representation contains all signals that belong to the signal group, regardless whether the application software has a data mapping defined or not.

As an example in figure 7.2 the 'Signal Group X' consists of the signals 'CRC', 'SC', 'D1', 'D2', 'D3', and 'D4'. Thus the RTE will interact with COM based on the whole array representation of this signal group with length of the signal group.

When the RTE interacts with the COM Based Transformer also all the other potential transformers need to be considered in order to determine which part of the array representation of the signal group actually is provided to each transformer since each transformer may add data during sending (or remove data during reception).

E.g. the part of the array representation which holds the 'CRC' and 'SC' signals are handled by the E2E transformer but will not be considered by the COM Based Transformer. The RTE will provide the data buffer from the 'Start of data with DataMapping defined' location to the COM Based Transformer.

7.1 Specification of the COM Based Transformer

Serialization describes the way data is represented in protocol data units (PDUs) transported over a network. For the COM Based Transformer the serialization is defined by the communication matrix using the System Template [7]. The communication matrix information is taken over to the Ecu configuration of the COM module.

[SWS_ComXf_00005] ⌈ The serialization is based on the Ecu configuration of the COM module and

- the Software Components `PortPrototype`,
- the `dataElement` list defined by the respective `SenderReceiverInterface`,
- the `SenderReceiverToSignalGroupMapping`,
- and the `ISignalToIPduMapping` for the `SystemSignalGroup`.

⌋*(SRS_Xfrm_00201, SRS_Xfrm_00202)*

[SWS_ComXf_00003] ⌈ The COM Based transformer shall only be used as the top-most transformer (first) in a transformer chain. ⌋*(SRS_Xfrm_00009, SRS_Xfrm_00011)*

The COM Based transformer serializes structured data into a linear form. Therefore it can only be used as the first transformer on the sending side and the last transformer on the receiving side.

[SWS_ComXf_00004] ⌈ The COM Based transformer defined in this document shall be used as a transformer if

- the attribute `protocol` of the `TransformationTechnology` is set to `COMBased`
- and the attribute `version` of the `TransformationTechnology` is set to `1`

⌋*(SRS_Xfrm_00201)*

[SWS_ComXf_00015] ⌈ The COM Based Transformer shall support all basic data types that are supported by the COM module in [SWS_Com_00675] except for `UINT8_DYN`. ⌋*(SRS_Xfrm_00201)*

[SWS_ComXf_00016] ⌈ The COM Based Transformer shall handle each `dataElement` of the `SenderReceiverInterface` individually. ⌋*(SRS_Xfrm_00201)*

[SWS_ComXf_00021] ⌈ The COM Based Transformer shall handle each `dataElement` of the `SenderReceiverInterface` like defined for the COM module [4] when the COM API `Com_SendSignal` (rep. `Com_ReceiveSignal`) is called for a shadow signal. ⌋*(SRS_Xfrm_00201)*

This defines that the COM Based Transformer performs all actions equally to the COM module. This does include functionality like endianess conversion and sign extension.

The COM Configuration implicitly defines the length of the signal group ([SWS_Com_00845] [4]) and the start position inside the ComIPdu where the signal group starts ([SWS_Com_00844] [4]). In order to place the transformed data element into the data buffer provided by the RTE the COM Based Transformer needs to respect the offset introduced by the position of the ComGroupSignal inside the ComIPdu (defined by the start position inside the ComIPdu) and the additional offset introduced by header data which is handled by other transformers called after the COM Based Transformer.

[SWS_ComXf_00014] ⌈ The COM Based Transformer shall calculate the 'COM Data Offset' by collecting all other transformer's headerLength information and add the start position inside the ComIPdu. ⌋*(SRS_Xfrm_00201, SRS_Xfrm_00202)*

[SWS_ComXf_00020] ⌈ The COM Based Transformer shall place the serialized data element into the data buffer at the position:
ComBitPosition - (8 * 'COM Data Offset')
where ComBitPosition is taken from the ComGroupSignal definition in COM and 'COM Data Offset' is defined by [SWS_ComXf_00014]. ⌋*(SRS_Xfrm_00201, SRS_Xfrm_00202)*

[SWS_ComXf_00013] ⌈ To allow migration, the deserialization shall be able to accept larger array representations and ignore dataElements appended at the end of a previously known parameter list. ⌋*(SRS_Xfrm_00201)*

This means: data elements that were not defined in the interface specification used to generate or parameterize the deserialization code at the end of the serialized data will be ignored by the deserialization.

Class	**TransformationTechnology**			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	A TransformationTechnology is a transformer inside a transformer chain. **Tags:** xml.namePlural=TRANSFORMATION-TECHNOLOGIES			
Base	ARObject,Identifiable,MultilanguageReferrable,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
bufferProperties	BufferProperties	1	aggr	Aggregation of the mandatory BufferProperties.
needsOriginalData	Boolean	0..1	attr	Specifies whether this transformer gets access to the SWC's original data.
protocol	String	1	attr	Specifies the protocol that is implemented by this transformer.
transformationDescription	TransformationDescription	0..1	aggr	A transformer can be configured with transformer specific parameters which are represented by the TransformerDescription. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
transformerClass	TransformerClassEnum	1	attr	Specifies to which transformer class this transformer belongs.
version	String	1	attr	Version of the implemented protocol.

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

Table 7.1: TransformationTechnology

Class	**BufferProperties**			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	Configuration of the buffer properties the transformer needs to work.			
Base	ARObject			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
bufferComputation	CompuScale	0..1	aggr	If the transformer changes the size of the data, the CompuScale can be used to specify a rule to derive the size of the output data based on the size of the input data.
headerLength	Integer	1	attr	Defines the length of the header (in bits) this transformer will add in front of the data.
inPlace	Boolean	1	attr	If set, the transformer uses the input buffer as output buffer.

Table 7.2: BufferProperties

Class	**TransformationDescription (abstract)**			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	The TransformationDescription is the abstract class that can be used by specific transformers to add transformer specific properties.			
Base	ARObject,Describable			
Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
–	–	–	–	–

Table 7.3: TransformationDescription

7.2 Error classififcation

7.2.1 Development Errors

[SWS_ComXf_00028] Development Errors ⌈ The errors shown in table 7.4 are defined as development errors for the COM Based Transformer. ⌋*(SRS_BSW_00337)*

Type of error	Related error code	Value
Error code if any other API service, except `GetVersionInfo` is called before the transformer module was initialized with `Init` or after a call to `DeInit`	`<MIP>_E_UNINIT`	0x01
Error code if an invalid configuration set was selected	`<MIP>_E_INIT_FAILED`	0x02
API service called with wrong parameter	`<MIP>_E_PARAM`	0x03
API service called with invalid pointer	`<MIP>_E_PARAM_POINTER`	0x04

Table 7.4: Development Errors

7.2.2 Runtime Errors

No runtime errors are specified for the COM Based Transformer.

7.2.3 Transient Faults

No transient faults are specified for the COM Based Transformer.

7.2.4 Production Errors

No production errors are specified for the COM Based Transformer.

7.2.5 Extended Production Errors

No extended production errors are specified for the COM Based Transformer.

8 API specification

8.1 Imported types

There are no imported types from other modules beyond those specified in [2].

In the Module Interlink Headers file which is imported by the COM Based Transformer, all `ImplementationDataType`s known to the RTE are included. Using this mechanism, the COM Based Transformer knows all data types of data which shall be transformed.

8.2 Type definitions

[SWS_ComXf_00030] ⌈

Name:	`ComXf_ConfigType`		
Type:	`Structure`		
Element:		`implementation specific`	–
Description:	This is the type of the data structure containing the initialization data for the transformer.		

Table 8.1: ComXf_ConfigType

⌋(*SRS_BSW_00404*, *SRS_BSW_00441*)

8.3 Function definitions

The COM Based transformer provides the specific interfaces generally required by [2].

[SWS_ComXf_00006] ⌈ The COM Based Transformer shall only provide functions for transformers where the `TransformationTechnology` is referenced as the first reference in the list of ordered references `transformer` from a `DataTransformation` to a `TransformationTechnology`. ⌋(*SRS_Xfrm_00011*)

That means, only the first transformer in a transformer chain can be a COM Based Transformer because serializer transformer are in general only allowed to be the first transformer in a chain.

8.3.1 ComXf_<transformerId>

[SWS_ComXf_00007] ⌈

Service name:	ComXf_<transformerId>

Syntax:	`uint8 ComXf_<transformerId>(`
	`uint8* buffer,`
	`uint16* bufferLength,`
	`const <type>* dataElement`
	`)`	

Service ID[hex]:	0x03	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	dataElement	Data element which shall be transformed
Parameters (inout):	None	
Parameters (out):	buffer bufferLength	Buffer allocated by the RTE, where the transformed data has to be stored by the transformer Used length of the buffer
Return value:	uint8	0x00 (E_OK): Serialization successful 0x81 (E_SER_GENERIC_ERROR): A generic error occurred
Description:	This function transforms a Sender/Receiver communication using the serialization of COM Based Transformer. It takes the data element as input and outputs an uint8 array containing the serialized data.	

Table 8.2: ComXf_<transformerId>

where

- `type` is data type of the data element
- `transformerId` is the name pattern for the transformer specified in [SWS_Xfrm_00062] ([2]).

⌋(*SRS_Xfrm_00201*)

The function specified in [SWS_ComXf_00007] exists for each transformed Sender/Receiver communication which uses the COM Based Transformer.

[SWS_ComXf_00008] ⌈ The function specified in [SWS_ComXf_00007] shall exist for the first reference in the list of ordered references `transformer` from a `DataTransformation` to a `TransformationTechnology` if the `DataTransformation` is referenced by an `ISignalGroup` in the role `comBasedSignalGroupTransformation` where the `ISignalGroup` references a `SystemSignalGroup` which is referenced by `SenderReceiverToSignalGroupMapping`. ⌋(*SRS_Xfrm_00201*)

[SWS_ComXf_00009] ⌈ The function specified in [SWS_ComXf_00007] shall serialize complex data elements of Sender/Receiver communication into a linear byte array representation using the COM Based Transformation. ⌋(*SRS_Xfrm_00201*)

8.3.2 ComXf_Inv_<transformerId>

[SWS_ComXf_00010] ⌈

Service name:	ComXf_Inv_<transformerId>

Syntax:	`uint8 ComXf_Inv_<transformerId>(`
	`const uint8* buffer,`
	`uint16 bufferLength,`
	`<type>* dataElement`
	`)`	

Service ID[hex]:	0x04	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	buffer bufferLength	Buffer allocated by the RTE, where the still serialized data are stored by the Rte Used length of the buffer
Parameters (inout):	None	
Parameters (out):	dataElement	Data element which is the result of the transformation and contains the deserialized data element
Return value:	uint8	0x00 (E_OK): Serialization successful 0x81 (E_SER_GENERIC_ERROR): A generic error occurred
Description:	This function deserializes a Sender/Receiver communication using the deserialization of COM Based Transformer. It takes the uint8 array containing the serialized data as input and outputs the original data element which will be passed to the Rte.	

Table 8.3: ComXf_Inv_<transformerId>

where

- `type` is data type of the data element
- `transformerId` is the name pattern for the transformer specified in [SWS_Xfrm_00062] ([2]).

⌋*(SRS_Xfrm_00201)*

The function specified in [SWS_ComXf_00010] exists for each transformed Sender/Receiver communication which uses the COM Based Transformation.

[SWS_ComXf_00011] ⌈ The function specified in [SWS_ComXf_00010] shall exist for the first reference in the list of ordered references `transformer` from a `DataTransformation` to a `TransformationTechnology` if the `DataTransformation` is referenced by an `ISignalGroup` in the role `comBasedSignalGroupTransformation` where the `ISignalGroup` references a `SystemSignalGroup` which is referenced by `SenderReceiverToSignalGroupMapping`. ⌋*(SRS_Xfrm_00201)*

[SWS_ComXf_00012] ⌈ The function specified in [SWS_ComXf_00010] shall deserialize a linear byte array to primitive or complex data elements of Sender/Receiver communication using the COM Based Transformation. ⌋*(SRS_Xfrm_00201)*

8.3.3 ComXf_Init

[SWS_ComXf_00026] ⌈

Service name:	ComXf_Init

Syntax:	`void ComXf_Init(`
	`const ComXf_ConfigType* config`
	`)`	

Service ID[hex]:	0x01	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	config	Pointer to the transformer's configuration data.
Parameters (inout):	None	
Parameters (out):	None	
Return value:	None	
Description:	This service initializes the transformer for the further processing.	

Table 8.4: ComXf_Init

⌋(*SRS_BSW_00407*, *SRS_BSW_00411*)

8.3.4 ComXf_DeInit

[SWS_ComXf_00027] ⌈

Service name:	ComXf_DeInit

Syntax:	`void ComXf_DeInit(`
	`void`
	`)`

Service ID[hex]:	0x02
Sync/Async:	Synchronous
Reentrancy:	Reentrant
Parameters (in):	None
Parameters (inout):	None
Parameters (out):	None
Return value:	None
Description:	This service deinitializes the transformer.

Table 8.5: ComXf_DeInit

⌋(*SRS_BSW_00407*, *SRS_BSW_00411*)

8.3.5 ComXf_GetVersionInfo

[SWS_ComXf_00024] ⌈

Service name:	ComXf_GetVersionInfo

Syntax:	`void ComXf_GetVersionInfo(`
	`Std_VersionInfoType* VersionInfo`
	`)`

Service ID[hex]:	0x00
Sync/Async:	Synchronous
Reentrancy:	Reentrant

Parameters (in):	None	
Parameters (inout):	None	
Parameters (out):	VersionInfo	Pointer to where to store the version information of this module.
Return value:	None	
Description:	This service returns the version information of the called transformer module.	

Table 8.6: ComXf_GetVersionInfo

⌋(*SRS_BSW_00407*, *SRS_BSW_00411*)

8.4 Callback notifications

COM Based Transformer has no callback notifications.

8.5 Scheduled functions

COM Based Transformer has no scheduled functions.

8.6 Expected interfaces

COM Based Transformer has no expected interfaces.

9 Sequence diagrams

There are no sequence diagrams applicable to COM Based Transformer.

10 Configuration specification

[SWS_ComXf_00031] ⌈ The COM Based Transformer is configured based on the COM module configuration [4]. ⌋*(SRS_Xfrm_00202)*

Still, there is an EcuC necessary to map the implementation of the transformer. The EcuC defined in [2] shall be used.

[SWS_ComXf_00025] ⌈ The `apiServicePrefix` of the COM Based Transformer's EcuC shall be set to `ComXf`. ⌋*(SRS_BSW_00159, SRS_Xfrm_00202)*

A Referenced Meta Classes

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	**DataTransformation**			
Package	M2::AUTOSARTemplates::SystemTemplate::Transformer			
Note	A DataTransformation represents a transformer chain. It is an ordered list of transformers.			
Base	ARObject,Identifiable,MultilanguageReferrable,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
executeDespiteDataUnavailability	Boolean	1	attr	Specifies whether the transformer is executed even if no input data are available.
transformerChain (ordered)	Transformation Technology	1..*	ref	

Table A.1: DataTransformation

Class	**EcucModuleDef**			
Package	M2::AUTOSARTemplates::ECUCParameterDefTemplate			
Note	Used as the top-level element for configuration definition for Software Modules, including BSW and RTE as well as ECU Infrastructure. **Tags:** atp.recommendedPackage=EcucModuleDefs			
Base	ARElement,ARObject,AtpBlueprint,AtpBlueprintable,AtpDefinition,CollectableElement,EcucDefinitionElement,Identifiable,MultilanguageReferrable,PackageableElement,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
apiServicePrefix	CIdentifier	0..1	ref	For CDD modules this attribute holds the apiServicePrefix. The shortName of the module definition of a Complex Driver is always "Cdd". Therefore for CDD modules the module apiServicePrefix is described with this attribute.
container	EcucContainerDef	1..*	aggr	Aggregates the top-level container definitions of this specific module definition. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=shortName xml.sequenceOffset=11
postBuildVariantSupport	Boolean	0..1	attr	Indicates if a module supports different post-build variants (previously known as post-build selectable configuration sets). TRUE means yes, FALSE means no.

Attribute	Datatype	Mul.	Kind	Note
refinedModuleDef	EcucModuleDef	0..1	ref	Optional reference from the Vendor Specific Module Definition to the Standardized Module Definition it refines. In case this EcucModuleDef has the category STANDARDIZED_MODULE_DEFINITION this reference shall not be provided. In case this EcucModuleDef has the category VENDOR_SPECIFIC_MODULE_DEFINITION this reference is mandatory. **Stereotypes:** atpUriDef
supportedConfigVariant	EcucConfigurationVariantEnum	*	attr	Specifies which ConfigurationVariants are supported by this software module. This attribute is optional if the EcucModuleDef has the category STANDARDIZED_MODULE_DEFINITION. If the category attribute of the EcucModuleDef is set to VENDOR_SPECIFIC_MODULE_DEFINITION then this attribute is mandatory.

Table A.2: EcucModuleDef

Class	ISignalGroup			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	SignalGroup of the Interaction Layer. The RTE supports a "signal fan-out" where the same System Signal Group is sent in different SignalIPdus to multiple receivers. An ISignalGroup refers to a set of ISignals that shall always be kept together. A ISignalGroup represents a COM Signal Group. Therefore it is recommended to put the ISignalGroup in the same Package as ISignals (see atp.recommendedPackage) **Tags:** atp.recommendedPackage=ISignalGroup			
Base	ARObject,CollectableElement,FibexElement,Identifiable,MultilanguageReferrable,PackageableElement,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
comBasedSignalGroupTransformation	DataTransformation	0..1	ref	Optional reference to a DataTransformation which represents the transformer chain that is used to transform the data that shall be placed inside this ISignalGroup based on the COMBasedTransformer approach. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=comBasedSignalGroupTransformation, variationPoint.shortLabel vh.latestBindingTime=codeGenerationTime
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.

Attribute	Datatype	Mul.	Kind	Note
transformationISignalProps	TransformationISignalProps	*	aggr	A transformer chain consists of an ordered list of transformers. The ISignalGroup specific configuration properties for each transformer are defined in the TransformationISignalProps class. The transformer configuration properties that are common for all ISignalGroups are described in the TransformationTechnology class.

Table A.3: ISignalGroup

Class	ISignalIPdu			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Represents the IPdus handled by Com. The ISignalIPdu assembled and disassembled in AUTOSAR COM consists of one or more signals. In case no multiplexing is performed this IPdu is routed to/from the Interface Layer. A maximum of one dynamic length signal per IPdu is allowed. **Tags:** atp.recommendedPackage=Pdus			
Base	ARObject,CollectableElement,FibexElement,IPdu,Identifiable,MultilanguageReferrable,PackageableElement,Pdu,Referrable			
Attribute	**Datatype**	**Mul.**	**Kind**	**Note**
iPduTimingSpecification	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

Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
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
unusedBitPattern	Integer	1	attr	AUTOSAR COM and AUTOSAR IPDUM are filling not used areas of an IPDU with this bit-pattern. This attribute is mandatory to avoid undefined behavior. This byte-pattern will be repeated throughout the IPdu.

Table A.4: ISignalIPdu

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*
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.

Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
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". The bit counting in byte 0 starts with bit 0 (least significant bit). The most significant bit in byte 0 is bit 7. Please note that the way the bytes will be actually sent on the bus does not impact this representation: they will always be seen by the software as a byte array. If a mapping for the ISignalGroup is defined, this attribute is irrelevant and shall be ignored.

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.

Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
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. This attribute denotes the least significant bit for "Little Endian" and the most significant bit for "Big Endian" packed signals within the IPdu (see the description of the packingByteOrder attribute). In AUTOSAR the bit counting is always set to "sawtooth" and the bit order is set to "Decreasing". The bit counting in byte 0 starts with bit 0 (least significant bit). The most significant bit in byte 0 is bit 7.

Table A.5: ISignalToIPduMapping

Class	**Implementation (abstract)**			
Package	M2::AUTOSARTemplates::CommonStructure::Implementation			
Note	Description of an implementation a single software component or module.			
Base	ARElement,ARObject,CollectableElement,Identifiable,MultilanguageReferrable,PackageableElement,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
buildActionManifest	BuildActionManifest	0..1	ref	A manifest specifying the intended build actions for the software delivered with this implementation. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=codeGenerationTime
codeDescriptor	Code	1..*	aggr	Specifies the provided implementation code.
compiler	Compiler	*	aggr	Specifies the compiler for which this implementation has been released

Attribute	*Datatype*	*Mul.*	*Kind*	*Note*
generated Artifact	DependencyOn Artifact	*	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	DependencyOn Artifact	*	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	DependencyOn Artifact	*	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. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=shortName
swVersion	RevisionLabelString	1	attr	Software version of this implementation. The numbering contains three levels (like major, minor, patch), its values are vendor specific.

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 A.6: 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*
dynamicArraySizeProfile	String	0..1	attr	Specifies the profile which the array will follow in case this data type is a variable size array.
subElement (ordered)	ImplementationDataTypeElement	*	aggr	Specifies an element of an array, struct, or union data type. The aggregation of ImplementionDataTypeElement is subject to variability with the purpose to support the conditional existence of elements inside a 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 A.7: ImplementationDataType

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 A.8: PortPrototype

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,Multilanguage Referrable,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 A.9: SenderReceiverInterface

Class	**SenderReceiverToSignalGroupMapping**			
Package	M2::AUTOSARTemplates::SystemTemplate::DataMapping			
Note	Mapping of a sender receiver communication data element with a composite datatype to a signal group.			
Base	ARObject,DataMapping			
Attribute	***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 A.10: SenderReceiverToSignalGroupMapping

Class	**SystemSignalGroup**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	A signal group refers to a set of signals that must always be kept together. A signal group is used to guarantee the atomic transfer of AUTOSAR composite data types. The SystemSignalGroup defines a signal grouping on VFB level. On cluster level the Signal grouping is described by the ISignalGroup element. **Tags:** atp.recommendedPackage=SystemSignalGroups			
Base	ARElement,ARObject,CollectableElement,Identifiable,MultilanguageReferrable,PackageableElement,Referrable			
Attribute	***Datatype***	***Mul.***	***Kind***	***Note***
systemSignal	SystemSignal	*	ref	Reference to a set of SystemSignals that must always be kept together.
transformingSystemSignal	SystemSignal	0..1	ref	Optional reference to the SystemSignal which shall contain the transformed (linear) data.

Table A.11: SystemSignalGroup

B Used ECU Configurtion

To ease the readability the relevant parts of the Com module configuration [4] are duplicated here (for information only).

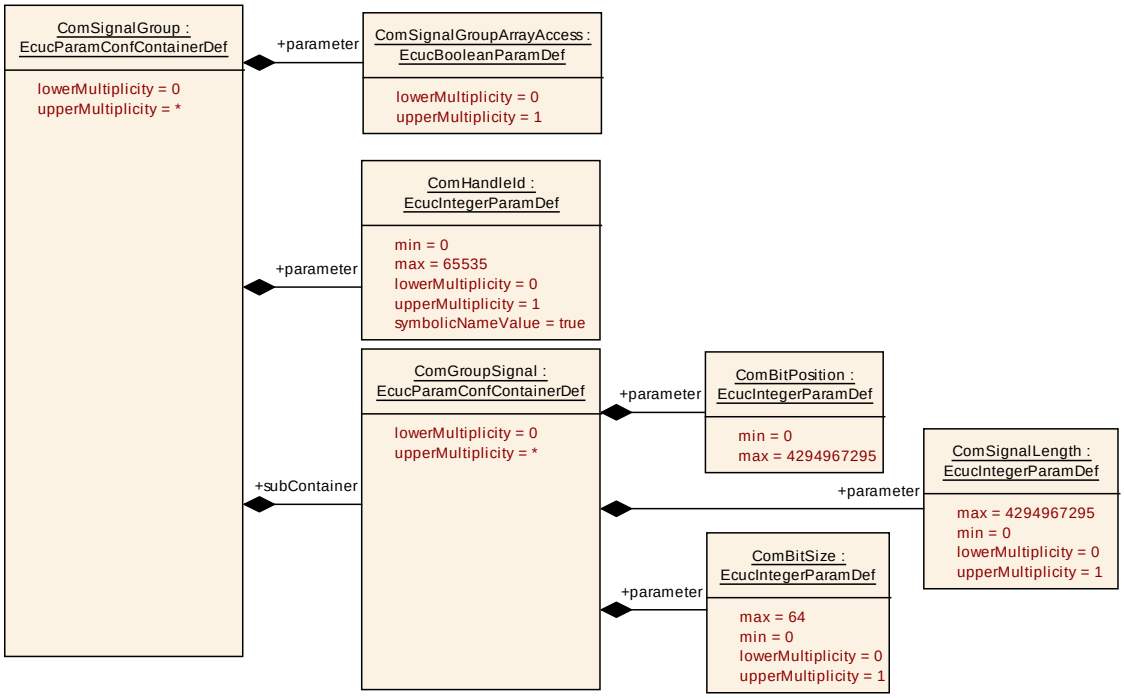

Figure B.1: AR_EcucDef_ComSignalGroupArrayAccess

ComSignalGroup

SWS Item	[ECUC_Com_00345]		
Container Name	ComSignalGroup		
Description	Contains the configuration parameters of the AUTOSAR COM module's signal groups.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

Name	ComDataInvalidAction [ECUC_Com_00314]		
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.		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	NOTIFY		
	REPLACE	Literal for DataInvalidAction	
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Scope / Dependency	scope: local		

Name	ComErrorNotification [ECUC_Com_00499]		
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.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Scope / Dependency	scope: local		

Name	ComFirstTimeout [ECUC_Com_00183]		
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.		
Multiplicity	0..1		
Type	EcucFloatParamDef		
Range	0 .. 3600		
Default Value			
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

Name	ComHandleId [ECUC_Com_00165]		
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.		
Multiplicity	0..1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	ComInitialValueOnly [ECUC_Com_00811]		
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).		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Scope / Dependency	scope: local		

Name	ComInvalidNotification [ECUC_Com_00315]		
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.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Scope / Dependency	scope: local		

Name	ComNotification [ECUC_Com_00498]		
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.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Scope / Dependency	scope: local		

Name	ComRxDataTimeoutAction [ECUC_Com_00412]		
Description	This parameter defines the action performed upon expiration of the reception deadline monitoring timer.		
Multiplicity	0..1		
Type	EcucEnumerationParamDef		
Range	NONE	no replacement shall take place	
	REPLACE	signals shall be replaced by their ComSignalInitValue	
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Scope / Dependency	scope: local		

Name	ComSignalGroupArrayAccess [ECUC_Com_10003]		
Description	Defines whether the uint8-array based access shall be used for this ComSignalGroup.		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency			

Name	ComSystemTemplateSignalGroupRef [ECUC_Com_00001]		
Description	Reference to the ISignalToIPduMapping that contains a reference to the ISignalGroup (SystemTemplate) which this ComSignalGroup represents.		
Multiplicity	0..1		
Type	Foreign reference to I-SIGNAL-TO-I-PDU-MAPPING		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

Name	ComTimeout [ECUC_Com_00263]	
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.	
Multiplicity	0..1	
Type	EcucFloatParamDef	
Range	0 .. 3600	
Default Value		
Post-Build Variant Multiplicity	true	

Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

Name	ComTimeoutNotification [ECUC_Com_00552]		
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.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Scope / Dependency	scope: local		

Name	ComTransferProperty [ECUC_Com_00232]	
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.	
Multiplicity	0..1	
Type	EcucEnumerationParamDef	
Range	PENDING	A write access to this signal never triggers the transmission of the corresponding I-PDU.
	TRIGGERED	Depending on the transmission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU.

	TRIGGERED_ON_CHANGE	Depending on the transmission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU, but only in case the written value is different to the locally stored (last sent or initial value) in length or value.	
	TRIGGERED_ON_CHANGE_WITHOUT_REPETITION	Depending on the transmission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU just once without a repetition, but only in case the written value is different to the locally stored (last sent or initial value) in length or value.	
	TRIGGERED_WITHOUT_REPETITION	Depending on the transmission mode, a write access to this signal can trigger the transmission of the corresponding I-PDU just once without a repetition.	
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

Name	ComUpdateBitPosition [ECUC_Com_00257]		
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..511 for CAN FD, 0..2031 for FlexRay, 0..4294967295 for TP.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default Value			
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD

Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
ComGroupSignal	0..*	This container contains the configuration parameters of group signals. I.e. signals that are included within a signal group.

ComGroupSignal

SWS Item	[ECUC_Com_00520]		
Container Name	ComGroupSignal		
Description	This container contains the configuration parameters of group signals. I.e. signals that are included within a signal group.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

Name	ComBitPosition [ECUC_Com_00259]		
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		
Multiplicity	1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default Value			
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

Name	ComBitSize [ECUC_Com_00158]		
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.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 64		
Default Value			
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

Name	ComHandleId [ECUC_Com_00165]		
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.		
Multiplicity	0..1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	ComSignalDataInvalidValue [ECUC_Com_00391]		
Description	Defines the data invalid value of the signal. In case the ComSignalType is UINT8, UINT16, UINT32, UINT64, SINT8, SINT16, SINT32, SINT64 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 interpreted as 0-sized dynamic signal.		
Multiplicity	0..1		
Type	EcucStringParamDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Scope / Dependency	scope: local dependency: In case of UINT8_N the length of ComSignalDataInvalidValue has to be the same as ComSignalLength.		

Name	ComSignalEndianness [ECUC_Com_00157]	
Description	Defines the endianness of the signal's network representation.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	BIG_ENDIAN	
	LITTLE_ENDIAN	
	OPAQUE	
Post-Build Variant Value	true	

Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

Name	ComSignalInitValue [ECUC_Com_00170]		
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, UINT64, SINT8, SINT16, SINT32, SINT64 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 interpreted as 0-sized dynamic signal.		
Multiplicity	0..1		
Type	EcucStringParamDef		
Default Value	0		
Regular Expression			
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local dependency: In case of UINT8_N the length of ComSignalInitValue has to be the same as ComSignalLength.		

Name	ComSignalLength [ECUC_Com_00437]		
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..64 for CAN FD I-PDUs, 0..254 for normal FlexRay I-PDUs (all of ComIPduType NORMAL), 0..4294967295 for I-PDUs with ComIPduType TP.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 4294967295		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Scope / Dependency	scope: local		

Name	ComSignalType [ECUC_Com_00127]	
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.	
Multiplicity	1	
Type	EcucEnumerationParamDef	
Range	BOOLEAN	
	FLOAT32	
	FLOAT64	
	SINT16	
	SINT32	
	SINT64	
	SINT8	
	UINT16	
	UINT32	
	UINT64	
	UINT8	
	UINT8_DYN	
	UINT8_N	
Post-Build Variant Value	false	

Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Scope / Dependency	scope: local		

Name	ComSystemTemplateSystemSignalRef [ECUC_Com_00002]		
Description	Reference to the ISignalToIPduMapping that contains a reference to the ISignal (System Template) which this ComSignal (or ComGroupSignal) represents.		
Multiplicity	0..1		
Type	Foreign reference to I-SIGNAL-TO-I-PDU-MAPPING		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU		

Name	ComTransferProperty [ECUC_Com_00560]	
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.	
Multiplicity	0..1	
Type	EcucEnumerationParamDef	
Range	PENDING	A change of the value of this group signal shall not be considered in the evaluation of the signal groups ComTransferProperty.
	TRIGGERED_ON_CHANGE	A change of the value of this group signal shall be considered in the evaluation of the signal groups ComTransferProperty.
Post-Build Variant Multiplicity	true	
Post-Build Variant Value	true	

Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
ComFilter	0..1	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.

ComIPdu

SWS Item	[ECUC_Com_00340]		
Container Name	ComIPdu		
Description	Contains the configuration parameters of the AUTOSAR COM module's I-PDUs.		
Post-Build Variant Multiplicity	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Configuration Parameters			

Name	ComIPduCallout [ECUC_Com_00387]		
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.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	

Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	ComIPduCancellationSupport [ECUC_Com_00709]		
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.		
Multiplicity	0..1		
Type	EcucBooleanParamDef		
Default Value	false		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	–	
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	–	
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: ECU dependency: This parameter shall not be set to true if ComCancellationSupport is set to false		

Name	ComIPduDirection [ECUC_Com_00493]		
Description	The direction defines if this I-PDU, and therefore the contributing signals and signal groups, shall be sent or received.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	RECEIVE		
	SEND		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Scope / Dependency	scope: local dependency: If configured to Sent also a ComTxIpdu container shall be included, see ECUC_Com_00496		

Name	ComIPduGroupRef [ECUC_Com_00206]		
Description	Reference to the I-PDU groups this I-PDU belongs to.		
Multiplicity	0..*		
Type	Reference to ComIPduGroup		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

Name	ComIPduHandleId [ECUC_Com_00175]		
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 or Com_TriggerIPDUSendWithMetaData, 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.		
Multiplicity	0..1		
Type	EcucIntegerParamDef (Symbolic Name generated for this parameter)		
Range	0 .. 65535		
Default Value			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: ECU		

Name	ComIPduSignalGroupRef [ECUC_Com_00519]		
Description	References to all signal groups contained in this I-Pdu		
Multiplicity	0..*		
Type	Reference to ComSignalGroup		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

Name	ComIPduSignalProcessing [ECUC_Com_00119]		
Description	For the definition of the two modes Immediate and Deferred.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	DEFERRED	signal indication / confirmations are deferred for example to a cyclic task	
	IMMEDIATE	the signal indications / confirmations are performed in Com_RxIndication/ Com_TxConfirmation	
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

Name	ComIPduSignalRef [ECUC_Com_00518]		
Description	References to all signals contained in this I-PDU.		
Multiplicity	0..*		
Type	Reference to ComSignal		
Post-Build Variant Multiplicity	true		
Post-Build Variant Value	true		
Multiplicity Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD

Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

Name	ComIPduTriggerTransmitCallout [ECUC_Com_00765]		
Description	If there is a trigger transmit callout defined for this I-PDU this parameter contains the name of the callout function.		
Multiplicity	0..1		
Type	EcucFunctionNameDef		
Default Value			
Regular Expression			
Post-Build Variant Multiplicity	false		
Post-Build Variant Value	false		
Multiplicity Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Value Configuration Class	**Pre-compile time**	X	All Variants
	Link time	–	
	Post-build time	–	
Scope / Dependency	scope: local		

Name	ComIPduType [ECUC_Com_00761]		
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.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	NORMAL	sent or received via normal L-PDU	
	TP	sent or received via TP	
Post-Build Variant Value	true		
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME
	Post-build time	X	VARIANT-POST-BUILD
Scope / Dependency	scope: local		

Name	ComPduIdRef [ECUC_Com_00711]		
Description	Reference to the "global" Pdu structure to allow harmonization of handle IDs in the COM-Stack.		
Multiplicity	1		
Type	Reference to Pdu		
Post-Build Variant Value	false		
Value Configuration Class	**Pre-compile time**	X	VARIANT-PRE-COMPILE
	Link time	X	VARIANT-LINK-TIME, VARIANT-POST-BUILD
	Post-build time	–	
Scope / Dependency			

Included Containers		
Container Name	**Multiplicity**	**Scope / Dependency**
ComIPduCounter	0..1	This optional container contains the configuration parameters of PDU Counter.
ComIPduReplication	0..1	This optional container contains the information needed for each I-PDU replicated.
ComTxIPdu	0..1	This container contains additional transmission related configuration parameters of the AUTOSAR COM module's I-PDUs.

C History of Constraints and Specification Items

C.1 Constraint History of this Document according to AUTOSAR R4.2.1

Initial document release.

C.2 Constraint History of this Document according to AUTOSAR R4.2.2

C.2.1 Added Traceables in 4.2.2

[SWS_ComXf_00032]

C.2.2 Changed Traceables in 4.2.2

[SWS_ComXf_00028]

C.2.3 Deleted Traceables in 4.2.2

none