

Document Title	SWS_SAEJ1939NetworkManagement: Complete Change Documentation 4.3.0 - 4.3.1
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	695

Document Status	Final
Part of AUTOSAR Standard	Classic Platform
Part of Standard Release	4.3.1

Table of Contents

1	SWS_SAEJ1939NetworkManagement	3
1.1	Specification Item ECUC_J1939Nm_00027	3
1.2	Specification Item ECUC_J1939Nm_00028	5
1.3	Specification Item SWS_J1939Nm_00028	8
1.4	Specification Item SWS_J1939Nm_00075	20

1 SWS_SAEJ1939NetworkManagement

1.1 Specification Item ECUC_J1939Nm_00027

Trace References:

none

Content:

Container Name	J1939NmConfigSetJ1939NmConfigSet
Description	This container is a MultipleConfigurationContainer, i.e. this container and its sub-containers exist once per configuration set contains the configuration parameters and sub containers of the AUTOSAR J1939Nm module.
Configuration Parameters	

Included parameters:

No Included Parameters

Included containers:

Included Containers		
Container Name	Multiplicity	Scope / Dependency
J1939NmChannel	1..*	Physical CAN channel handled by J1939Nm.
J1939NmExternalNode	0..*	Logical node implemented in another ECU. Configures potential communication partners. If this container is connected to more than one channel, the external ECU is linked to the local ECU by each of these channels.
J1939NmNode	1..*	Logical node representing one function handled by J1939Nm.
J1939NmSharedAddressSpace	0..*	Set of J1939NmChannels that share a common address space. Address claims will be routed between these channels.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77386: MultipleConfigurationContainer occurrences

Problem description:

As far as I know, MultipleConfigurationContainer type has been removed from the standard.

However, there are some SWS documents which mention MultipleConfigura-

tionContainer.

COM:

Figure 10: ComConfig: "multipleConfigurationContainer = true"

ECUC_Com_00540: "This container is a MultipleConfigurationContainer, i.e. this container and its sub-containers exist once per configuration set."

J1939Nm:

ECUC_J1939Nm_00027, ECUC_J1939Nm_00028: "This container is a Multiple-ConfigurationContainer, i.e. this container and its sub-containers exist once per configuration set."

PduR:

Figure 33: ComConfig: "multipleConfigurationContainer = true", CanIfInitCfg: "multipleConfigurationContainer = true"

Rte:

Chapter 5.3.10: "RtePostBuildVariantConfiguration is a multipleConfigurationContainer"

Agreed solution:

COM:

Remove multipleConfigurationContainer from (description of) ComConfig and regenerate and update chap10/Com.html and Figure 10 (The AUTOSAR COM modules Configuration Overview).

=====

Rte:

- In chapter 5.3.10 remove block "RtePostBuildVariantConfiguration is a multiple-ConfigurationContainer... post build configurable inside the RTE."

- In chapter 5.3.10.3 replace "And likewise for the example 2 header file the RTE generator can declare and initialize in the Rte_PBcfg.c file all possible Post-BuildVariantCriterionValueSets and the RtePostBuildVariantConfigurations using references to these variant sets." with "And likewise for the example 2 header file the RTE generator can declare and initialize in the Rte_PBcfg.c file all possible PostBuildVariantCriterionValueSets and the RtePostBuildVariantConfiguration using references to these variant sets."

- In chapter 7.4 replace "Each instance of this container specifies one Post-Build variant of the generated Rte. The shortName of the container RtePost-BuildVariantConfiguration specifies the variant name." with "Each instance of

RtePostBuildUsedPredefinedVariant inside this container specifies one PostBuild variant of the generated Rte. The shortName of the RtePostBuildUsedPredefinedVariant specifies the variant name."

- [ECUC_Rte_09084]: Remove "The shortName of this container defines the name of the RtePostBuildVariant."

- [ECUC_Rte_09083]: Add "The shortName of the referenced PredefinedVariant defines the name of the RtePostBuildVariant."

=====

J1939Nm:

Change the description of J1939NmConfigSet (ECUC_J1939Nm_00027) to: "This container contains the configuration parameters and sub containers of the AUTOSAR J1939Nm module."

=====

PduR:

Remove multipleConfigurationContainer from ComConfig and CanIfInitConfig on Figure 29.

=====

–Last change on issue 77386 comment 8–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.2 Specification Item ECUC_J1939Nm_00028

Trace References:

none

Content:

Module Name	J1939NmJ1939Nm
Module Description	Configuration of the J1939 Network Management module.
Post-Build Variant Support	true
Supported Config Variants	VARIANT-LINK-TIME, VARIANT-POST-BUILD, VARIANT-PRE-COMPILE

Included containers:

Included Containers		
Container Name	Multiplicity	Scope / Dependency
J1939NmConfigSet	1	This container is a MultipleConfiguration Container, i.e. this container and its sub-containers exist once per configuration set. It contains the configuration parameters and sub-containers of the AUTOSAR J1939Nm module.
J1939NmGeneral	1	Contains the general configuration parameters of the module.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77386: MultipleConfigurationContainer occurrences

Problem description:

As far as I know, MultipleConfigurationContainer type has been removed from the standard.

However, there are some SWS documents which mention MultipleConfigurationContainer.

COM:

Figure 10: ComConfig: "multipleConfigurationContainer = true"

ECUC_Com_00540: "This container is a MultipleConfigurationContainer, i.e. this container and its sub-containers exist once per configuration set."

J1939Nm:

ECUC_J1939Nm_00027, ECUC_J1939Nm_00028: "This container is a Multiple-ConfigurationContainer, i.e. this container and its sub-containers exist once per configuration set."

PduR:

Figure 33: ComConfig: "multipleConfigurationContainer = true", CanIfInitCfg: "multipleConfigurationContainer = true"

Rte:

Chapter 5.3.10: "RtePostBuildVariantConfiguration is a multipleConfigurationContainer"

Agreed solution:

COM:

Remove multipleConfigurationContainer from (description of) ComConfig and

regenerate and update chap10/Com.html and Figure 10 (The AUTOSAR COM modules Configuration Overview).

=====

Rte:

- In chapter 5.3.10 remove block "RtePostBuildVariantConfiguration is a multiple-ConfigurationContainer... post build configurable inside the RTE."

- In chapter 5.3.10.3 replace "And likewise for the example 2 header file the RTE generator can declare and initialize in the Rte_PBcfg.c file all possible PostBuildVariantCriterionValueSets and the RtePostBuildVariantConfigurations using references to these variant sets." with "And likewise for the example 2 header file the RTE generator can declare and initialize in the Rte_PBcfg.c file all possible PostBuildVariantCriterionValueSets and the RtePostBuildVariantConfiguration using references to these variant sets."

- In chapter 7.4 replace "Each instance of this container specifies one Post-Build variant of the generated Rte. The shortName of the container RtePostBuildVariantConfiguration specifies the variant name." with "Each instance of RtePostBuildUsedPredefinedVariant inside this container specifies one PostBuild variant of the generated Rte. The shortName of the RtePostBuildUsedPredefinedVariant specifies the variant name."

- [ECUC_Rte_09084]: Remove "The shortName of this container defines the name of the RtePostBuildVariant."

- [ECUC_Rte_09083]: Add "The shortName of the referenced PredefinedVariant defines the name of the RtePostBuildVariant."

=====

J1939Nm:

Change the description of J1939NmConfigSet (ECUC_J1939Nm_00027) to: "This container contains the configuration parameters and sub containers of the AUTOSAR J1939Nm module."

=====

PduR:

Remove multipleConfigurationContainer from ComConfig and CanIfInitConfig on Figure 29.

=====
 –Last change on issue 77386 comment 8–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.3 Specification Item SWS_J1939Nm_00028

Trace References:

none

Content:

Service name:	< User_AddressClaimedIndication >User_AddressClaimedIndication	
Syntax:	void < User_AddressClaimedIndication >(NetworkHandleType channel, uint8 sourceAddress, const uint8* name)	
Service ID[hex]:	0x20	
Sync/Async:	Synchronous	
Reentrancy:	Reentrant	
Parameters (in):	channelUser_AddressClaimed Indication.channel	Channel on which the AC was received.
	sourceAddressUser_AddressClaimed Indication.sourceAddress	Address of the node that sent the AC or NULL address (0xFE).
	nameUser_AddressClaimed Indication.name	Pointer to the byte array containing the 64bit NAME.
Parameters (inout):	None	
Parameters (out):	None	
Return value:	None	
Description:	Provides the content of received AddressClaimed (AC) PGs.	

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #68035: [diverse] Introduce rules defining which input parameters shall be passed per value and which ones per const reference

Problem description:

SWS_BSW_00186 especially states that input pointer parameters shall use the const qualifier (i.e., shall be P2CONST).

In addition to that there shall be a SWS item that states that input parameters

of integral and enum type shall be passed by value whereas input parameters of structure type shall be passed by reference.

The various transformer SWS documents shall be adapted accordingly.
–Last change on issue 68035 comment 4–

Agreed solution:

BSW UML model

The attachment "Changed Proposal in WP-A meeting" contains a list of changes to the APIs in the model (see column H). Afterwards all related documents (included in impact list) shall update their generated artifacts.

General Requirements on Basic Software Modules

~~~~~

Introduce the following requirements prior to SRS\_BSW\_00371:

SRS\_BSW\_XXXX: Input parameters of scalar and enum types shall be passed as a value.

Type: valid

Description: All input parameters of scalar or enum type shall be passed as a value.

Rationale:

Use case: For example a function named <Mip>\_SomeFunction with a return type of Std\_ReturnType and a single parameter named SomeParameter of type uint8 is defined with the following signature:

Std\_ReturnType <Mip>\_SomeFunction(uint8 SomeParameter);

Dependencies: –

Supporting Material: —

SRS\_BSW\_YYYY: Input parameters of structure type shall be passed as a reference to a constant structure

Type: valid

Description: All input parameters of structure type shall be passed as a reference constant structure

Rationale: Passing input parameters of structure type by value would result in additional run-time overhead due to efforts for copying the whole structure.

Use case: For example a function named `<Mip>_SomeFunction` with a return type of `Std_ReturnType` and a single parameter named `SomeParameter` of type `SomeStructure` (where `SomeStructure` is a struct) is defined with the following signature:

```
Std_ReturnType <Mip>_SomeFunction(P2CONST(SomeStructure, AUTOMATIC,
<MIP>_APPL_DATA) SomeParameter);
```

Dependencies: –

Supporting Material: —

**SRS\_BSW\_zzzzz:** Input parameters of array type shall be passed as a reference to the constant array base type

Type: valid

Description: All input parameters of array type shall be passed as a reference to the constant array base type

Rationale: This effectively matches the behavior specified in the ISO-C:90 namely that a "declaration of a parameter as 'array of type' shall be adjusted to 'qualified pointer to type'".

Use case: For example a function named `<Mip>_SomeFunction` with a return type of `Std_ReturnType` and a single parameter named `SomeParameter` of type array of `uint8` is defined with the following signature:

```
Std_ReturnType <Mip>_SomeFunction(P2CONST(uint8, AUTOMATIC,
<MIP>_APPL_DATA) SomeParameter);
```

Dependencies: –

Supporting Material: —

### General Specification of Transformers

~~~~~

In SWS_Xfrm_00036 change

`const <type>* dataElement`

to

`<paramtype> dataElement`

and add the following to the where clause after the API table after the bullet "type is data type of the data element"

"

<paramtype> is derived from <type> according to the parameter passing rules rules defined by the SRS BSW General (see SRS_BSW_xxxxx, SRS_BSW_yyyyy, and SRS_BSW_zzzzz) and SWS BSW General (see SWS_BSW_00186 and SWS_BSW_00187).

In SWS_Xfrm_00038 change

[<type> data_1,] ...
[<type> data_n]

to

[<paramtype> data_1,] ...
[<paramtype> data_n]

and add the following to the where clause after the API table after the bullet "type is data type of the data element"

"

<paramtype> is derived from <type> according to the parameter passing rules rules defined by the SRS BSW General (see SRS_BSW_xxxxx, SRS_BSW_yyyyy, and SRS_BSW_zzzzz) and SWS BSW General (see SWS_BSW_00186 and SWS_BSW_00187).

The following paragraph shall then be removed:

For the arguments of ClientServerOperation which are handed over to the transformer as data_1, ..., data_n the requirements to API parameters stated in chapter API Parameters of [5, SWS RTE] are valid (especially [SWS_Rte_01017], [SWS_Rte_01018] and [SWS_Rte_05107]).

In SWS_Xfrm_00040 change

[<originalData1>, ...
<originalDataN>]

to

[<paramtype> originalData1,] ...
[<paramtype> originalDataN]

and add the following to the where clause after the API table after the bullet
"type is data type of the data element
"

<paramtype> is derived from <type> according to the parameter passing rules
rules defined by the SRS BSW General (see SRS_BSW_xxxxx, SRS_BSW_yyyyy,
and SRS_BSW_zzzzz) and SWS BSW General (see SWS_BSW_00186 and
SWS_BSW_00187).

In SWS_Xfrm_00044 change

<type> *data_1, ...
<type> *data_n

to

[<paramtype> data_1,] ...
[<paramtype> data_n]

and add the following to the where clause after the API table after the bullet
"type is data type of the data element
"

<paramtype> is derived from <type> according to the parameter passing rules
rules defined by the SRS BSW General (see SRS_BSW_xxxxx, SRS_BSW_yyyyy,
and SRS_BSW_zzzzz) and SWS BSW General (see SWS_BSW_00186 and
SWS_BSW_00187).

The following paragraph shall then be removed:

For the arguments of ClientServerOperation which are handed over to the
transformer as data_1, ..., data_n the requirements to API parameters stated in
chapter API Parameters of [5, SWS RTE] are valid (especially [SWS_Rte_01017],
[SWS_Rte_01018] and [SWS_Rte_05107]).

Speci?cation of SOME/IP Transformer
~~~~~

In SWS\_SomelpXf\_00138 change

const <type>\* dataElement

to

<paramtype> dataElement

and add the following to the where clause after the API table after the bullet  
"type is data type of the data element  
"

<paramtype> is derived from <type> according to the parameter passing rules  
rules defined by the SRS BSW General (see SRS\_BSW\_xxxxx, SRS\_BSW\_yyyyy,  
and SRS\_BSW\_zzzzz) and SWS BSW General (see SWS\_BSW\_00186 and  
SWS\_BSW\_00187).

In SWS\_SomelpXf\_00141 change

[<type> data\_1,] ...

[<type> data\_n]

to

[<paramtype> data\_1,] ...

[<paramtype> data\_n]

and add the following to the where clause after the API table after the bullet  
"type is data type of the data element  
"

<paramtype> is derived from <type> according to the parameter passing rules  
rules defined by the SRS BSW General (see SRS\_BSW\_xxxxx, SRS\_BSW\_yyyyy,  
and SRS\_BSW\_zzzzz) and SWS BSW General (see SWS\_BSW\_00186 and  
SWS\_BSW\_00187).

The following paragraph shall then be removed:

For the arguments of ClientServerOperation which are handed over to the  
transformer as data\_1, ..., data\_n the requirements to API parameters stated in

chapter API Parameters of [5, SWS RTE] are valid (especially [SWS\_Rte\_01017], [SWS\_Rte\_01018] and [SWS\_Rte\_05107]).

In SWS\_SomelpXf\_00145 change

<type> \*data\_1, ...  
<type> \*data\_n

to

[<paramtype> data\_1,] ...  
[<paramtype> data\_n]

and add the following to the where clause after the API table after the bullet "type is data type of the data element"

<paramtype> is derived from <type> according to the parameter passing rules rules defined by the SRS BSW General (see SRS\_BSW\_xxxxx, SRS\_BSW\_yyyyy, and SRS\_BSW\_zzzzz) and SWS BSW General (see SWS\_BSW\_00186 and SWS\_BSW\_00187).

The following paragraph shall then be removed:

For the arguments of ClientServerOperation which are handed over to the transformer as data\_1, ..., data\_n the requirements to API parameters stated in chapter API Parameters of [5, SWS RTE] are valid (especially [SWS\_Rte\_01017], [SWS\_Rte\_01018] and [SWS\_Rte\_05107]).

### Specification of COM Based Transformer

~~~~~

In SWS_ComXf_00007 change

const <type>* dataElement

to

<paramtype> dataElement

and add the following to the where clause after the API table after the bullet
"type is data type of the data element
"

<paramtype> is derived from <type> according to the parameter passing rules
rules defined by the SRS BSW General (see SRS_BSW_xxxxx, SRS_BSW_yyyyy,
and SRS_BSW_zzzzz) and SWS BSW General (see SWS_BSW_00186 and
SWS_BSW_00187).

Specification of Time Sync over Ethernet

~~~~~

In SWS\_EthTSyn\_00040 make the parameter DataPtr of EthTSyn\_RxIndication  
const.

### Specification of SWS FlexRay Interface

~~~~~

Change SWS_Frlf_05073 from
Frlf_NumOfStartupFramesPtr (IN)
to
Frlf_NumOfStartupFramesPtr (OUT)

Specification of ADC

~~~~~

~[SWS\_Adc\_00419] Adc\_SetupResultBuffer: change Adc\_ValueGroupType\* to  
const Adc\_ValueGroupType\*  
~[SWS\_Adc\_00369] Adc\_ReadGroup: move Adc\_ValueGroupType \* from Parame-  
ters (in) to Parameters (out)

There is no need to change parameter from IN to INOUT in Adc\_SetupResultBuffer

### Specification of Com

~~~~~

Change type of parameter MetaData of Com_TriggerIPDUSendWithMetaData from
uint8* to const uint8*

Specification of ComM

~~~~~

no change required

Specification of Dem

~~~~~

no change required

Specification of DLT

~~~~~

no change required

Specification of DoIP

~~~~~

From:

Std_ReturnType <User>_DoIPRoutingActivationConfirmation(boolean* Confirmed,
uint8* ConfirmationReqData, uint8* ConfirmationResData)

Std_ReturnType <User>_DoIPRoutingActivationAuthentication(boolean* Authenti-
fied, uint8* AuthenticationReqData, uint8* AuthenticationResData)

To:

Std_ReturnType <User>_DoIPRoutingActivationConfirmation(boolean* Confirmed,
const uint8* ConfirmationReqData, uint8* ConfirmationResData)

Std_ReturnType <User>_DoIPRoutingActivationAuthentication(boolean* Authenti-
fied, const uint8* AuthenticationReqData, uint8* AuthenticationResData)

Specification of E2ELibrary

~~~~~

no change required

Specification of Eth

~~~~~

no change required

Specification of EthIf

~~~~~

no change required

### Specification of EthSwitchDriver

~~~~~

no change required

Specification of ICUDriver

~~~~~

SWS\_Icu\_00201: Icu\_StartTimestamp

Parameter (IN): Icu\_ValueType\* BufferPtr shall be changed to Parameters (out) type

### Specification of LdCom

~~~~~

[SWS_LDCOM_00027]: LdCom_CopyTxData

BufReq_ReturnType LdCom_CopyTxData(PduIdType id, const PduInfoType* info, RetryInfoType* retry, PduLengthType* availableDataPtr) shall be changed to
BufReq_ReturnType LdCom_CopyTxData(PduIdType id, const PduInfoType* info, const RetryInfoType* retry, PduLengthType* availableDataPtr)

[SWS_LDCOM_00036]: Rte_LdComCbkJCopyTxData_<sn>

BufReq_ReturnType Rte_LdComCbkJCopyTxData_<sn>(const PduInfoType* info, RetryInfoType* retry, PduLengthType* availableDataPtr) shall be changed to
BufReq_ReturnType Rte_LdComCbkJCopyTxData_<sn>(const PduInfoType* info, const RetryInfoType* retry, PduLengthType* availableDataPtr)

Specification of Lin

~~~~~

PduInfoPtr needs to be const in Std\_ReturnType Lin\_SendFrame( uint8 Channel, const Lin\_PduType\* PduInfoPtr )

### Specification of PduR

~~~~~

* PduR_<User:LoTp>CopyTxData
add const to "RetryInfoType* retry"

Specification of J1939Nm

~~~~~

Change parameter 'name' of User\_AddressClaimedIndication to type 'const uint8\*\*'

#### Specification of SoAd

~~~~~

=> everything already fixed with RfC 65633

Specification of SPIHandlerDriver

~~~~~

==> nothing to change for SWS SPI

#### Specification of SynchronizedTimeBaseManager

~~~~~

"StbM not affected. All issues listed in the WP-A attachment have been already implemented by IT 69124 in context of RfC 65633"

Specification of Tcplp

~~~~~

~[SWS\_TCPIP\_00040] Tcplp\_DhcpReadOption: change DataPtr from (IN) to (OUT)

~[SWS\_TCPIP\_00189] Tcplp\_DhcpV6ReadOption: change DataPtr from (IN) to (OUT)

=> everything else already fixed with RfC 65633

#### Specification of TimeSyncOverFlexRay

~~~~~

"Change SWS_FrTSyn_00064: parameter versioninfo of type Std_VersionInfoType* is marked wrongly as IN. Change to OUT"

Specification of EFX

~~~~~

~ [SWS\_Efx\_00355] Efx\_Debounce\_u8\_u8: Include constant for pointer Input-parameter as like below.

uint8 Efx\_Debounce\_u8\_u8( boolean X, Efx\_DebounceState\_Type \* State, const Efx\_DebounceParam\_Type \* Param, sint32 dT )

~ [SWS\_Efx\_00376] Efx\_MedianSort: The parameter <InType>\* Array should be InOut instead of In parameter as like below.

Parameters (in): N Size of an array  
 Parameters (inout): Array Pointer to an array

~ [SWS\_Efx\_00309] Efx\_RampCheckActivity: Include constant for pointer Input-parameter as like below.

boolean Efx\_RampCheckActivity(const Efx\_StateRamp\_Type\* State\_cpst)

~ [SWS\_Efx\_00307] Efx\_RampGetSwitchPos: Include constant for pointer Input-parameter as like below.

boolean Efx\_RampGetSwitchPos(const Efx\_StateRamp\_Type\* State\_cpst)

~ [SWS\_Efx\_00193] Efx\_Array\_Average: Include constant for pointer Input-parameter as like below.

<OutType> Efx\_Array\_Average\_<InTypeMn>\_<OutTypeMn>( const <InType>\* Array, uint16 Count)

#### Specification of MFL

~~~~~

~ [SWS_Mfl_00192] Mfl_Debounce_u8_u8: Include constant for pointer Input-parameter as like below.

boolean Mfl_Debounce_u8_u8(boolean X, Mfl_DebounceState_Type* State, const Mfl_DebounceParam_Type* Param, float32 dT)

~ [SWS_Mfl_00266] Mfl_DebounceInit: The parameter Mfl_DebounceState_Type* State should be Out instead of In parameter as like below.

Parameters (in): X Initial value for the input state

Parameters (out): State Pointer to structure for debouncing state variables

~ [SWS_Mfl_00246] Mfl_HystDeltaRight_f32_u8: Include constant for pointer Input-parameter as like below.

boolean Mfl_HystDeltaRight_f32_u8(float32 X, float32 Delta, float32 Rsp, const uint8* State)

~ [SWS_Mfl_00285] Mfl_MedianSort_f32_f32: The parameter Array should be InOut instead of In parameter as like below.

Parameters (in): N Size of an array

Parameters (inout): Array Pointer to an array

~ [SWS_Mfl_00037] Mfl_PT1SetState: The parameter State_cpst should be Out instead of In parameter as like below.

Parameters (in): X1_f32 Initial value for input state

Y1_f32 Initial value for output state

Parameters (out): State_cpst Pointer to internal state structure

~ [SWS_Mfl_00225] Mfl_RampCheckActivity: Include constant for pointer Input-parameter as like below.

boolean Mfl_RampCheckActivity(const Mfl_StateRamp_Type* State_cpst)

~ [SWS_Mfl_00223] Mfl_RampGetSwitchPos: Include constant for pointer Input-parameter as like below.

boolean Mfl_RampGetSwitchPos(const Mfl_StateRamp_Type* State_cpst)

–Last change on issue 68035 comment 135–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.4 Specification Item SWS_J1939Nm_00075

Trace References:

[SRS_BSW_00415](#)

Content:

The implementation (J1939Nm.c) shall include the file BswM_J1939Nm.h.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76875: [J1939Nm] Minor documentation bugs

Problem description:

There are minor documentation issues in J1939Nm module SWS, check them below

Issue 1:

Page 9 of AR J1939Nm SWS under the section 4.2 Applicability to car domains

Below statement looks misleading,

J1939 is developed by the SAE as a standard for heavy-duty on-highway, farming, and construction vehicles. It is not applicable to passenger cars or light trucks. The ****J1939 Request Manager**** will for now only be used in heavy-duty on-highway vehicles, because other domains are currently excluded by AUTOSAR.

Clarification:

The part enclosed with ** part must either be AR J1939 or AR J1939NM module. It is not required to mention J1939RM inside this section of SWS of J1939Nm.

Issue 2:

Figure 1 in Page 11, there is a modelling element SchM_1939Nm.h

Clarification:

This has to be SchM_J1939Nm.h

Issue 3:

J1939Nm_RxIndication and J1939Nm_TxConfirmation are called by CanIf.

Under requirement tables [SWS_J1939Nm_00037] and [SWS_J1939Nm_00036], the description refers to CanIf as lower layer communication interface module

Clarification:

As J1939 is only over CAN, we can replace this description with CanIf.

Issue 4:

On Page 42, Under J1939NmTxpdu, the SWS says Contains the configuration of the I-PDU used to transmit the AddressClaimed PG.

Clarification:

This is wrong, this Pdu can be used to send both the AC and Cannot Claim address PG

Same has to be changed in ECUC_J1939Nm_00009

Issue 5:

J1939Nm needs an interface from BSWM, it is not mentioned which file needs to be included to see the declaration of the function.

Clarification:

This needs to be mentioned in Chapter 5

Agreed solution:

BSW UML Model:

In the include hierarchy diagram of J1939:

- add BswM_J1939Nm.h and

- replace SchM_1939Nm.h by SchM_J1939Nm.h.

=====

SWS J1939Nm:

Add a new requirement to section 5.1.2:

[SWS_J1939Nm_xxxx1] /The implementation (J1939Nm.c) shall include the file BswM_J1939Nm.h./ (SRS_BSW_00415)
–Last change on issue 76875 comment 11–

BW-C-Level:

Application	Specification	Bus
1	1	1