Document Title	SWS_IPDUMultiplexer: 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_IPDUMultiplexer

1.1 Specification Item ECUC_IpduM_00059

Trace References:

none

Content:

Container Name	IpduMConfig
Description	This container contains the sub containers of the IpduM module. * The IpduMTxPathway subcontainer includes information about sent I-PDUs. * The IpduMRxPathway includes information about received I-PDUs. * The IpduMTxContainerMContainerTxPdu and IpduMTx ContainedMContainedTxPdu include information about the sending of ContainerPdus. * The IpduMRxContainerMContainerRxPdu and IpduMRxContainedMContainedRxPdu include information about the reception of ContainerPdus.
Configuration Parameters	

Included parameters:

Included Parameters	
Parameter Name	SWS Item ID
IpduMMaxTxBufferSize	ECUC_IpduM_00166
IpduMMaxTxPathwayCnt	ECUC_IpduM_00165

Included containers:

Included Containers		
Container Name	Multiplicity	Scope / Dependency
IpduMContainedRxPdu	0..*	Configuration of a received contained Pdu.
IpduMContainedTxPdu	0..*	Configuration of a sender Contained Pdu.
IpduMContainerRxPdu	0..*	Configuration of a receiver Container Pdu which may collect several ContainedPdus.
IpduMContainerTxPdu	0..*	Configuration of a transmitted container Pdu.
IpduMRxPathway	0..*	includes information about received I-PDUs
IpduMTxPathway	0..*	includes information about sent I-PDUs

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73303: Misspelled container names in descriptions of configuration

Problem description:

The chapter ’10 Configuration specification’ provides misspelled names of containers and references in descriptions.
This comprises the wrong names IpduMRxContainerPdu, IpduMTxContainerPdu, IpduMRxContainedPdu,IpduMTxContainedPdu and IpduMRxContainedPduContainerRef.

Agreed solution:

Correct all occurrences of
IpduMRxContainerPdu by IpduMContainerRxPdu,
IpduMTxContainerPdu by IpduMContainerTxPdu,
IpduMRxContainedPdu by IpduMContainedRxPdu,
IpduMTxContainedPdu by IpduMContainedTxPdu and
IpduMRxContainedPduContainerRef by IpduMContainedRxInContainerPduRef.

BW-C-Level:

Application	Specification	Bus
1	1	1

1.2 Specification Item ECUC_IpduM_00172

Trace References:

none

Content:

Name	IpduMContainedPduHeaderIdIpduMContainedTxPdu.IpduMContainedPduHeaderId		
Parent Container	IpduMContainedTxPdu		
Description	Header Id which is part of the ContainerPdu when this ContainedPdu is inside.		
Multiplicity	1 0..1		
Type	EcucIntegerParamDef		
Range	1 .. 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 dependency: only valid if IpduMContainerHeaderSize is set to IPDUM_HEADERTYPE_SHORT or IPDUM_HEADERTYPE_LONG.		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76543: [IpduM] Container Pdu not usable for CAN-FD

  Problem description:

  With the CAN-FD extension, the Container PDU concept has been introduced. Unfortunately this it not usable in a meaningful way for CAN-FD due to the 4-byte header for each contained PDU of a Container Pdu.

  Agreed solution:

  Proposed solution for the SWS IPduM:

  Extend the enumeration of IpduMContainerHeaderSize by:
  - IPDUM_HEADERTYPE_LONG
  - IPDUM_HEADERTYPE_SHORT
  - IPDUM_HEADERTYPE_NONE

  The detailed proposed solution is contained here: https://svn.autosar.org/repos/work/09_WorkPackages/WP-A2/70_InternalDocuments/RfCs/76543-AUTOSAR_SWS_IPDUMultiplexer.doc

  ==========
  Proposed solution TPS_SystemTemplate:

  Extend the enumeration of ContainerIPduHeaderTypeEnum by:
  - noHeader: No Header is used and the the location of each containedPdu in the ContainerPdu is statically configured.

  Add the following optional attributes to ContainedIPduProps
  - offset (PositiveInteger): Byte offset that describes the location of the ContainedPdu in the ContainerPdu if no header is used.
  - updateIndicationBitPosition (PositiveInteger): The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated.

  Add the following optional attributes to ContainerIPdu
  - unusedBitPattern (PositiveInteger): IPduM fills not updated areas of the ContainerPdu with this byte-pattern.

  Rewrite the Introduction of chapter 6.3.1 ContainerIPdu

  For each IPdu which is put inside a ContainerIPdu, a header may be provided which

determines which IPdu is contained (ContainedIPduProps.headerIdLongHeader or headerIdShortHeader and what the size of that IPdu is (DLC during runtime). With this header mode the receivers are able to extract the individual contained IPdus again. As an alternative option to the usage of headers a statically configured layout of IPdus in the ContainerIPdu is supported.

Introduce the following constraints:

[constr_xxxx1] Mandatory offset if noHeader is used
For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu.containedPduTriggering with ContainerIPdu.headerType = noHeader the IPdu.containedIPduProps.offset shall be defined.

[constr_xxxx2] Usage of ContainerIPdu.rxAcceptContainedIPdu if noHeader is used
If the ContainerIPdu.headerType is set to noHeader then the ContainerIPdu.rxAcceptContainedIPdu attribute value shall be set to acceptConfigured.

[constr_xxxx3] Usage of ContainedIPduProps.updateIndicationBitPosition
ContainedIPduProps.updateIndicationBitPosition is only allowed to be set to a value if the headerType of the ContainerIPdu that contains the IPdu with containedIPduProps is set to noHeader.

[constr_xxxx4] Only the last contained IPdu (according to the ContainedIPduProps.offset) of a ContainerIPdu with static container layout (i.e., a ContainerIPdu with ContainerIPduHeaderTypeEnum set to noHeader) may be a dynamic length PDU (i.e, a PDU that at runtime may exhibit a length different from the one statically configured via Pdu.length of the respective Pdu). All other contained PDUs of a ContainerIPdu with static container layout have to be static length PDUs.

Adapt the following spec items to:

[TPS_SYST_02098] Header id and header type of a contained IPdu
A contained IPdu shall always have the same headerId per header type (long or short header), regardless in which ContainerIPdu it is collected. If noHeader is set then the contained Ipdu does not need to have a headerId.

[TPS_SYST_02100] Relation between ContainerIPdu and contained IPdus on receiver side
On receiver side, it is not necessarily required to statically define which IPdus may be contained inside a ContainerIPdu if the header mode is used. Thus it would be possible to update the senders of ContainerIPdus and put different or additional

IPdus inside.

Rework descriptions of the following classes and attributes in the model:

ContainerIPdu class description: Allows to collect several IPdus in one ContainerIPdu based on the headerType
ContainerIPdu.headerType attribute description: Defines whether and which header type is used (header id and length).

==========
Proposed solution SRS_IpduM:

- rename Chapter 6.1.4 to Dynamic I-PDU to Container Mapping
rename [SRS_IpduM_02820] to Dynamic I-PDU Mapping
add a dependency: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_0XXXX) but no mixture.
- add new Chapter Static I-PDU to Container Mapping
- add new SRS requirement: Static I-PDU Mapping
Type: Draft
Description: Multiple I-PDUs shall be mappable to a static position into a Container-PDU. If an I-PDU is mapped statically to a Container it is always transported in the same position of the Container-PDU.
Rationale: Support scenarios of Container Mapping where no header shall be used to reduce bandwidth in case always all contained PDU shall be sent.
Use Case: Efficient transmission of small PDUs on high bandwidth busses
Dependencies: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_02820) but no mixture.
–Last change on issue 76543 comment 100–

BW-C-Level:

Application	Specification	Bus
1	2	1

1.3 Specification Item ECUC_IpduM_00174

Trace References:

none

Content:

Container Name	IpduMContainedRxPduIpduMContainedRxPdu		
Description	Configuration of a received contained Pdu.		
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			

Included parameters:

Included Parameters	
Parameter Name	SWS Item ID
IpduMContainedPduOffset	ECUC_IpduM_00206
IpduMContainedRxPduLongHeaderId	ECUC_IpduM_00203
IpduMContainedRxPduShortHeaderId	ECUC_IpduM_00202
IpduMPduUpdateBitPosition	ECUC_IpduM_00207
IpduMContainedRxInContainerPduRef	ECUC_IpduM_00173
IpduMContainedRxPduRef	ECUC_IpduM_00175

Included containers:

No Included Containers

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76543: [IpduM] Container Pdu not usable for CAN-FD

  Problem description:

  With the CAN-FD extension, the Container PDU concept has been introduced. Unfortunately this it not usable in a meaningful way for CAN-FD due to the 4-byte header for each contained PDU of a Container Pdu.

  Agreed solution:

  Proposed solution for the SWS IPduM:

  Extend the enumeration of IpduMContainerHeaderSize by:
  - IPDUM_HEADERTYPE_LONG
  - IPDUM_HEADERTYPE_SHORT
  - IPDUM_HEADERTYPE_NONE

The detailed proposed solution is contained here: https://svn.autosar.org/repos/work/09_WorkPackages/WP-A2/70_InternalDocuments/RfCs/76543-AUTOSAR_SWS_IPDUMultiplexer.doc

==========
Proposed solution TPS_SystemTemplate:

Extend the enumeration of ContainerIPduHeaderTypeEnum by:
- noHeader: No Header is used and the the location of each containedPdu in the ContainerPdu is statically configured.

Add the following optional attributes to ContainedIPduProps
- offset (PositiveInteger): Byte offset that describes the location of the ContainedPdu in the ContainerPdu if no header is used.
- updateIndicationBitPosition (PositiveInteger): The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated.

Add the following optional attributes to ContainerIPdu
- unusedBitPattern (PositiveInteger): IPduM fills not updated areas of the ContainerPdu with this byte-pattern.

Rewrite the Introduction of chapter 6.3.1 ContainerIPdu

For each IPdu which is put inside a ContainerIPdu, a header may be provided which determines which IPdu is contained (ContainedIPduProps.headerIdLongHeader or headerIdShortHeader and what the size of that IPdu is (DLC during runtime). With this header mode the receivers are able to extract the individual contained IPdus again. As an alternative option to the usage of headers a statically configured layout of IPdus in the ContainerIPdu is supported.

Introduce the following constraints:

[constr_xxxx1] Mandatory offset if noHeader is used
For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu.containedPduTriggering with ContainerIPdu.headerType = noHeader the IPdu.containedIPduProps.offset shall be defined.

[constr_xxxx2] Usage of ContainerIPdu.rxAcceptContainedIPdu if noHeader is used

If the ContainerIPdu.headerType is set to noHeader then the ContainerIPdu.rxAcceptContainedIPdu attribute value shall be set to acceptConfigured.

[constr_xxxx3] Usage of ContainedIPduProps.updateIndicationBitPosition
ContainedIPduProps.updateIndicationBitPosition is only allowed to be set to a value if the headerType of the ContainerIPdu that contains the IPdu with containedIPduProps is set to noHeader.

[constr_xxxx4] Only the last contained IPdu (according to the ContainedIPduProps.offset) of a ContainerIPdu with static container layout (i.e., a ContainerIPdu with ContainerIPduHeaderTypeEnum set to noHeader) may be a dynamic length PDU (i.e, a PDU that at runtime may exhibit a length different from the one statically configured via Pdu.length of the respective Pdu). All other contained PDUs of a ContainerIPdu with static container layout have to be static length PDUs.

Adapt the following spec items to:

[TPS_SYST_02098] Header id and header type of a contained IPdu
A contained IPdu shall always have the same headerId per header type (long or short header), regardless in which ContainerIPdu it is collected. If noHeader is set then the contained Ipdu does not need to have a headerId.

[TPS_SYST_02100] Relation between ContainerIPdu and contained IPdus on receiver side
On receiver side, it is not necessarily required to statically define which IPdus may be contained inside a ContainerIPdu if the header mode is used. Thus it would be possible to update the senders of ContainerIPdus and put different or additional IPdus inside.

Rework descriptions of the following classes and attributes in the model:

ContainerIPdu class description: Allows to collect several IPdus in one ContainerIPdu based on the headerType
ContainerIPdu.headerType attribute description: Defines whether and which header type is used (header id and length).

==========
Proposed solution SRS_IpduM:

- rename Chapter 6.1.4 to Dynamic I-PDU to Container Mapping

rename [SRS_IpduM_02820] to Dynamic I-PDU Mapping
add a dependency: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_0XXXX) but no mixture.
- add new Chapter Static I-PDU to Container Mapping
- add new SRS requirement: Static I-PDU Mapping
Type: Draft
Description: Multiple I-PDUs shall be mappable to a static position into a Container-PDU. If an I-PDU is mapped statically to a Container it is always transported in the same position of the Container-PDU.
Rationale: Support scenarios of Container Mapping where no header shall be used to reduce bandwidth in case always all contained PDU shall be sent.
Use Case: Efficient transmission of small PDUs on high bandwidth busses
Dependencies: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_02820) but no mixture.
–Last change on issue 76543 comment 100–

BW-C-Level:

Application	Specification	Bus
1	2	1

1.4 Specification Item ECUC_IpduM_00177

Trace References:

none

Content:

Container Name	IpduMContainedTxPduIpduMContainedTxPdu		
Description	Configuration of a sender ContainedPdu.		
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			

Included parameters:

Included Parameters	
Parameter Name	SWS Item ID
IpduMContainedPduHeaderId	ECUC_IpduM_00172
IpduMContainedPduOffset	ECUC_IpduM_00206

Included Parameters	
Parameter Name	SWS Item ID
IpduMContainedTxPduCollectionSemantics	ECUC_IpduM_00198
IpduMContainedTxPduConfirmation	ECUC_IpduM_00178
IpduMContainedTxPduHandleId	ECUC_IpduM_00179
IpduMContainedTxPduSendTimeout	ECUC_IpduM_00181
IpduMContainedTxPduTrigger	ECUC_IpduM_00182
IpduMPduUpdateBitPosition	ECUC_IpduM_00207
IpduMContainedTxInContainerPduRef	ECUC_IpduM_00176
IpduMContainedTxPduRef	ECUC_IpduM_00180

Included containers:

No Included Containers

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76543: [IpduM] Container Pdu not usable for CAN-FD

  Problem description:

  With the CAN-FD extension, the Container PDU concept has been introduced. Unfortunately this it not usable in a meaningful way for CAN-FD due to the 4-byte header for each contained PDU of a Container Pdu.

  Agreed solution:

  Proposed solution for the SWS IPduM:

  Extend the enumeration of IpduMContainerHeaderSize by:
  - IPDUM_HEADERTYPE_LONG
  - IPDUM_HEADERTYPE_SHORT
  - IPDUM_HEADERTYPE_NONE

  The detailed proposed solution is contained here: https://svn.autosar.org/repos/work/09_WorkPackages/WP-A2/70_InternalDocuments/RfCs/76543-AUTOSAR_SWS_IPDUMultiplexer.doc

  ==========

  Proposed solution TPS_SystemTemplate:

Extend the enumeration of ContainerIPduHeaderTypeEnum by:
- noHeader: No Header is used and the the location of each containedPdu in the ContainerPdu is statically configured.

Add the following optional attributes to ContainedIPduProps
- offset (PositiveInteger): Byte offset that describes the location of the ContainedPdu in the ContainerPdu if no header is used.
- updateIndicationBitPosition (PositiveInteger): The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated.

Add the following optional attributes to ContainerIPdu
- unusedBitPattern (PositiveInteger): IPduM fills not updated areas of the ContainerPdu with this byte-pattern.

Rewrite the Introduction of chapter 6.3.1 ContainerIPdu

For each IPdu which is put inside a ContainerIPdu, a header may be provided which determines which IPdu is contained (ContainedIPduProps.headerIdLongHeader or headerIdShortHeader and what the size of that IPdu is (DLC during runtime). With this header mode the receivers are able to extract the individual contained IPdus again. As an alternative option to the usage of headers a statically configured layout of IPdus in the ContainerIPdu is supported.

Introduce the following constraints:

[constr_xxxx1] Mandatory offset if noHeader is used
For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu.containedPduTriggering with ContainerIPdu.headerType = noHeader the IPdu.containedIPduProps.offset shall be defined.

[constr_xxxx2] Usage of ContainerIPdu.rxAcceptContainedIPdu if noHeader is used
If the ContainerIPdu.headerType is set to noHeader then the ContainerIPdu.rxAcceptContainedIPdu attribute value shall be set to acceptConfigured.

[constr_xxxx3] Usage of ContainedIPduProps.updateIndicationBitPosition
ContainedIPduProps.updateIndicationBitPosition is only allowed to be set to a value if the headerType of the ContainerIPdu that contains the IPdu with containedIPduProps is set to noHeader.

[constr_xxxx4] Only the last contained IPdu (according to the Con-

tainedIPduProps.offset) of a ContainerIPdu with static container layout (i.e., a ContainerIPdu with ContainerIPduHeaderTypeEnum set to noHeader) may be a dynamic length PDU (i.e, a PDU that at runtime may exhibit a length different from the one statically configured via Pdu.length of the respective Pdu). All other contained PDUs of a ContainerIPdu with static container layout have to be static length PDUs.

Adapt the following spec items to:

[TPS_SYST_02098] Header id and header type of a contained IPdu
A contained IPdu shall always have the same headerId per header type (long or short header), regardless in which ContainerIPdu it is collected. If noHeader is set then the contained Ipdu does not need to have a headerId.

[TPS_SYST_02100] Relation between ContainerIPdu and contained IPdus on receiver side
On receiver side, it is not necessarily required to statically define which IPdus may be contained inside a ContainerIPdu if the header mode is used. Thus it would be possible to update the senders of ContainerIPdus and put different or additional IPdus inside.

Rework descriptions of the following classes and attributes in the model:

ContainerIPdu class description: Allows to collect several IPdus in one ContainerIPdu based on the headerType
ContainerIPdu.headerType attribute description: Defines whether and which header type is used (header id and length).

==========
Proposed solution SRS_IpduM:

- rename Chapter 6.1.4 to Dynamic I-PDU to Container Mapping
rename [SRS_IpduM_02820] to Dynamic I-PDU Mapping
add a dependency: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_0XXXX) but no mixture.
- add new Chapter Static I-PDU to Container Mapping
- add new SRS requirement: Static I-PDU Mapping
Type: Draft
Description: Multiple I-PDUs shall be mappable to a static position into a Container-PDU. If an I-PDU is mapped statically to a Container it is always transported in the same position of the Container-PDU.

Rationale: Support scenarios of Container Mapping where no header shall be used to reduce bandwidth in case always all contained PDU shall be sent.
Use Case: Efficient transmission of small PDUs on high bandwidth busses
Dependencies: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_02820) but no mixture.
–Last change on issue 76543 comment 100–

BW-C-Level:

Application	Specification	Bus
1	2	1

1.5 Specification Item ECUC_IpduM_00183

Trace References:

none

Content:

Name	IpduMContainerHeaderSizeIpduMContainerTxPdu.IpduMContainerHeaderSize		
Parent Container	IpduMContainerTxPdu		
Description	Defines the layout of the header information (header id and length).		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	IPDUM_HEADERTYPE_LONG IpduMContainerTxPdu.Ipdu MContainerHeader Size.IPDUM_HEADERTYPE_LONG	Header size is 64 bit: * Header Id 32 bit * Dlc 32 bit	
	IPDUM_HEADERTYPE_NONE IpduMContainerTxPdu.Ipdu MContainerHeader Size.IPDUM_HEADERTYPE_NONE	Static Container Layout Tags: atp.Status=draft	
	IPDUM_HEADERTYPE_SHORT IpduMContainerTxPdu.Ipdu MContainerHeader Size.IPDUM_HEADERTYPE_SHORT	Header size is 32 bit: * Header Id 24 bit * Dlc 8 bit	
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		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76543: [IpduM] Container Pdu not usable for CAN-FD

Problem description:

With the CAN-FD extension, the Container PDU concept has been introduced. Unfortunately this it not usable in a meaningful way for CAN-FD due to the 4-byte header for each contained PDU of a Container Pdu.

Agreed solution:

Proposed solution for the SWS IPduM:

Extend the enumeration of IpduMContainerHeaderSize by:
- IPDUM_HEADERTYPE_LONG
- IPDUM_HEADERTYPE_SHORT
- IPDUM_HEADERTYPE_NONE

The detailed proposed solution is contained here: https://svn.autosar.org/repos/work/09_WorkPackages/WP-A2/70_InternalDocuments/RfCs/76543-AUTOSAR_SWS_IPDUMultiplexer.doc

==========
Proposed solution TPS_SystemTemplate:

Extend the enumeration of ContainerIPduHeaderTypeEnum by:
- noHeader: No Header is used and the the location of each containedPdu in the ContainerPdu is statically configured.

Add the following optional attributes to ContainedIPduProps
- offset (PositiveInteger): Byte offset that describes the location of the ContainedPdu in the ContainerPdu if no header is used.
- updateIndicationBitPosition (PositiveInteger): The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated.

Add the following optional attributes to ContainerIPdu
- unusedBitPattern (PositiveInteger): IPduM fills not updated areas of the ContainerPdu with this byte-pattern.

Rewrite the Introduction of chapter 6.3.1 ContainerIPdu

For each IPdu which is put inside a ContainerIPdu, a header may be provided which determines which IPdu is contained (ContainedIPduProps.headerIdLongHeader or

headerIdShortHeader and what the size of that IPdu is (DLC during runtime). With this header mode the receivers are able to extract the individual contained IPdus again. As an alternative option to the usage of headers a statically configured layout of IPdus in the ContainerIPdu is supported.

Introduce the following constraints:

[constr_xxxx1] Mandatory offset if noHeader is used
For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu.containedPduTriggering with ContainerIPdu.headerType = noHeader the IPdu.containedIPduProps.offset shall be defined.

[constr_xxxx2] Usage of ContainerIPdu.rxAcceptContainedIPdu if noHeader is used
If the ContainerIPdu.headerType is set to noHeader then the ContainerIPdu.rxAcceptContainedIPdu attribute value shall be set to acceptConfigured.

[constr_xxxx3] Usage of ContainedIPduProps.updateIndicationBitPosition
ContainedIPduProps.updateIndicationBitPosition is only allowed to be set to a value if the headerType of the ContainerIPdu that contains the IPdu with containedIPduProps is set to noHeader.

[constr_xxxx4] Only the last contained IPdu (according to the ContainedIPduProps.offset) of a ContainerIPdu with static container layout (i.e., a ContainerIPdu with ContainerIPduHeaderTypeEnum set to noHeader) may be a dynamic length PDU (i.e, a PDU that at runtime may exhibit a length different from the one statically configured via Pdu.length of the respective Pdu). All other contained PDUs of a ContainerIPdu with static container layout have to be static length PDUs.

Adapt the following spec items to:

[TPS_SYST_02098] Header id and header type of a contained IPdu
A contained IPdu shall always have the same headerId per header type (long or short header), regardless in which ContainerIPdu it is collected. If noHeader is set then the contained Ipdu does not need to have a headerId.

[TPS_SYST_02100] Relation between ContainerIPdu and contained IPdus on receiver side
On receiver side, it is not necessarily required to statically define which IPdus may be contained inside a ContainerIPdu if the header mode is used. Thus it would be possible to update the senders of ContainerIPdus and put different or additional IPdus inside.

Rework descriptions of the following classes and attributes in the model:

ContainerIPdu class description: Allows to collect several IPdus in one ContainerIPdu based on the headerType
ContainerIPdu.headerType attribute description: Defines whether and which header type is used (header id and length).

==========
Proposed solution SRS_IpduM:

- rename Chapter 6.1.4 to Dynamic I-PDU to Container Mapping
rename [SRS_IpduM_02820] to Dynamic I-PDU Mapping
add a dependency: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_0XXXX) but no mixture.
- add new Chapter Static I-PDU to Container Mapping
- add new SRS requirement: Static I-PDU Mapping
Type: Draft
Description: Multiple I-PDUs shall be mappable to a static position into a Container-PDU. If an I-PDU is mapped statically to a Container it is always transported in the same position of the Container-PDU.
Rationale: Support scenarios of Container Mapping where no header shall be used to reduce bandwidth in case always all contained PDU shall be sent.
Use Case: Efficient transmission of small PDUs on high bandwidth busses
Dependencies: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_02820) but no mixture.
–Last change on issue 76543 comment 100–

BW-C-Level:

Application	Specification	Bus
1	2	1

1.6 Specification Item ECUC_IpduM_00186

Trace References:

none

Content:

Name	IpduMContainerRxAcceptContainedPdu

Description	Defines for the received IpduMRxContainerMContainerRxPdu whether the list of referencing IpduMRxContainedMContainedRxPdus (via the reference IpduMRxContainedMContainedPdu ContainerRefRx) is a closed set.		
Multiplicity	1		
Type	EcucEnumerationParamDef		
Range	IPDUM_ACCEPT_ALL	The IpduMRxContainedMContainedRxPdus which are referencing this IpduMRxContainerMContainerRxPdu are expected inside this IpduMRxContainerMContainerRxPdu, but there may also occur other Pdus inside this IpduMRx ContainerMContainerRxPdu as well. This also supports the case where no IpduMRxContainedMContainedRxPdu references the IpduMRxContainerMContainerRxPdu.	
	IPDUM_ACCEPT_CONFIGURED	Only the IpduMRxContainedMContainedRxPdus which are referencing this IpduMRxContainerMContainerRxPdu are expected inside this IpduMRxContainerMContainerRxPdu.	
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		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73303: Misspelled container names in descriptions of configuration

  Problem description:

  The chapter ’10 Configuration specification’ provides misspelled names of containers and references in descriptions.
  This comprises the wrong names IpduMRxContainerPdu, IpduMTxContainerPdu, IpduMRxContainedPdu,IpduMTxContainedPdu and IpduMRxContainedPduContainerRef.

  Agreed solution:

  Correct all occurrences of
  IpduMRxContainerPdu by IpduMContainerRxPdu,
  IpduMTxContainerPdu by IpduMContainerTxPdu,
  IpduMRxContainedPdu by IpduMContainedRxPdu,
  IpduMTxContainedPdu by IpduMContainedTxPdu and
  IpduMRxContainedPduContainerRef by IpduMContainedRxInContainerPduRef.

  BW-C-Level:

Application	Specification	Bus
1	1	1

1.7 Specification Item ECUC_IpduM_00192

Trace References:

none

Content:

Container Name	IpduMContainerTxPduIpduMContainerTxPdu		
Description	Configuration of a transmitted container Pdu.		
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			

Included parameters:

Included Parameters	
Parameter Name	SWS Item ID
IpduMContainerHeaderSize	ECUC_IpduM_00183
IpduMContainerQueueSize	ECUC_IpduM_00185
IpduMContainerTxFirstContainedPduTrigger	ECUC_IpduM_00199
IpduMContainerTxHandleId	ECUC_IpduM_00191
IpduMContainerTxSendTimeout	ECUC_IpduM_00194
IpduMContainerTxSizeThreshold	ECUC_IpduM_00195
IpduMContainerTxTriggerMode	ECUC_IpduM_00196
IpduMUnusedAreasDefault	ECUC_IpduM_00208
IpduMContainerTxPduRef	ECUC_IpduM_00193

Included containers:

No Included Containers

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76543: [IpduM] Container Pdu not usable for CAN-FD

  Problem description:

  With the CAN-FD extension, the Container PDU concept has been introduced. Unfortunately this it not usable in a meaningful way for CAN-FD due to the 4-byte header for each contained PDU of a Container Pdu.

Agreed solution:

Proposed solution for the SWS IPduM:

Extend the enumeration of IpduMContainerHeaderSize by:
- IPDUM_HEADERTYPE_LONG
- IPDUM_HEADERTYPE_SHORT
- IPDUM_HEADERTYPE_NONE

The detailed proposed solution is contained here: https://svn.autosar.org/repos/work/09_WorkPackages/WP-A2/70_InternalDocuments/RfCs/76543-AUTOSAR_SWS_IPDUMultiplexer.doc

==========
Proposed solution TPS_SystemTemplate:

Extend the enumeration of ContainerIPduHeaderTypeEnum by:
- noHeader: No Header is used and the the location of each containedPdu in the ContainerPdu is statically configured.

Add the following optional attributes to ContainedIPduProps
- offset (PositiveInteger): Byte offset that describes the location of the ContainedPdu in the ContainerPdu if no header is used.
- updateIndicationBitPosition (PositiveInteger): The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated.

Add the following optional attributes to ContainerIPdu
- unusedBitPattern (PositiveInteger): IPduM fills not updated areas of the ContainerPdu with this byte-pattern.

Rewrite the Introduction of chapter 6.3.1 ContainerIPdu

For each IPdu which is put inside a ContainerIPdu, a header may be provided which determines which IPdu is contained (ContainedIPduProps.headerIdLongHeader or headerIdShortHeader and what the size of that IPdu is (DLC during runtime). With this header mode the receivers are able to extract the individual contained IPdus again. As an alternative option to the usage of headers a statically configured layout of IPdus in the ContainerIPdu is supported.

Introduce the following constraints:

[constr_xxxx1] Mandatory offset if noHeader is used
For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu.containedPduTriggering with ContainerIPdu.headerType = noHeader the IPdu.containedIPduProps.offset shall be defined.

[constr_xxxx2] Usage of ContainerIPdu.rxAcceptContainedIPdu if noHeader is used
If the ContainerIPdu.headerType is set to noHeader then the ContainerIPdu.rxAcceptContainedIPdu attribute value shall be set to acceptConfigured.

[constr_xxxx3] Usage of ContainedIPduProps.updateIndicationBitPosition
ContainedIPduProps.updateIndicationBitPosition is only allowed to be set to a value if the headerType of the ContainerIPdu that contains the IPdu with containedIPduProps is set to noHeader.

[constr_xxxx4] Only the last contained IPdu (according to the ContainedIPduProps.offset) of a ContainerIPdu with static container layout (i.e., a ContainerIPdu with ContainerIPduHeaderTypeEnum set to noHeader) may be a dynamic length PDU (i.e, a PDU that at runtime may exhibit a length different from the one statically configured via Pdu.length of the respective Pdu). All other contained PDUs of a ContainerIPdu with static container layout have to be static length PDUs.

Adapt the following spec items to:

[TPS_SYST_02098] Header id and header type of a contained IPdu
A contained IPdu shall always have the same headerId per header type (long or short header), regardless in which ContainerIPdu it is collected. If noHeader is set then the contained Ipdu does not need to have a headerId.

[TPS_SYST_02100] Relation between ContainerIPdu and contained IPdus on receiver side
On receiver side, it is not necessarily required to statically define which IPdus may be contained inside a ContainerIPdu if the header mode is used. Thus it would be possible to update the senders of ContainerIPdus and put different or additional IPdus inside.

Rework descriptions of the following classes and attributes in the model:

ContainerIPdu class description: Allows to collect several IPdus in one Con-

tainerIPdu based on the headerType
ContainerIPdu.headerType attribute description: Defines whether and which header type is used (header id and length).

==========
Proposed solution SRS_IpduM:

- rename Chapter 6.1.4 to Dynamic I-PDU to Container Mapping
rename [SRS_IpduM_02820] to Dynamic I-PDU Mapping
add a dependency: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_0XXXX) but no mixture.
- add new Chapter Static I-PDU to Container Mapping
- add new SRS requirement: Static I-PDU Mapping
Type: Draft
Description: Multiple I-PDUs shall be mappable to a static position into a Container-PDU. If an I-PDU is mapped statically to a Container it is always transported in the same position of the Container-PDU.
Rationale: Support scenarios of Container Mapping where no header shall be used to reduce bandwidth in case always all contained PDU shall be sent.
Use Case: Efficient transmission of small PDUs on high bandwidth busses
Dependencies: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_02820) but no mixture.
–Last change on issue 76543 comment 100–

BW-C-Level:

Application	Specification	Bus
1	2	1

1.8 Specification Item ECUC_IpduM_00195

Trace References:

none

Content:

Name	IpduMContainerTxSizeThresholdIpduMContainerTxPdu.IpduMContainerTxSizeThreshold
Parent Container	IpduMContainerTxPdu
Description	Defines the size threshold in bytes which, when exceeded, triggers the sending of the Container Pdu although the maxium Pdu size (PduLength parameter of Pdu object) has not been reached yet.
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 dependency: only valid if IpduMContainerHeaderSize is set to IPDUM_HEADERTYPE_SHORT or IPDUM_HEADERTYPE_LONG		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76543: [IpduM] Container Pdu not usable for CAN-FD

  Problem description:

  With the CAN-FD extension, the Container PDU concept has been introduced. Unfortunately this it not usable in a meaningful way for CAN-FD due to the 4-byte header for each contained PDU of a Container Pdu.

  Agreed solution:

  Proposed solution for the SWS IPduM:

  Extend the enumeration of IpduMContainerHeaderSize by:
  - IPDUM_HEADERTYPE_LONG
  - IPDUM_HEADERTYPE_SHORT
  - IPDUM_HEADERTYPE_NONE

  The detailed proposed solution is contained here: https://svn.autosar.org/repos/work/09_WorkPackages/WP-A2/70_InternalDocuments/RfCs/76543-AUTOSAR_SWS_IPDUMultiplexer.doc

  ==========
  Proposed solution TPS_SystemTemplate:

  Extend the enumeration of ContainerIPduHeaderTypeEnum by:
  - noHeader: No Header is used and the the location of each containedPdu in the

ContainerPdu is statically configured.

Add the following optional attributes to ContainedIPduProps
- offset (PositiveInteger): Byte offset that describes the location of the ContainedPdu in the ContainerPdu if no header is used.
- updateIndicationBitPosition (PositiveInteger): The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated.

Add the following optional attributes to ContainerIPdu
- unusedBitPattern (PositiveInteger): IPduM fills not updated areas of the ContainerPdu with this byte-pattern.

Rewrite the Introduction of chapter 6.3.1 ContainerIPdu

For each IPdu which is put inside a ContainerIPdu, a header may be provided which determines which IPdu is contained (ContainedIPduProps.headerIdLongHeader or headerIdShortHeader and what the size of that IPdu is (DLC during runtime). With this header mode the receivers are able to extract the individual contained IPdus again. As an alternative option to the usage of headers a statically configured layout of IPdus in the ContainerIPdu is supported.

Introduce the following constraints:

[constr_xxxx1] Mandatory offset if noHeader is used
For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu.containedPduTriggering with ContainerIPdu.headerType = noHeader the IPdu.containedIPduProps.offset shall be defined.

[constr_xxxx2] Usage of ContainerIPdu.rxAcceptContainedIPdu if noHeader is used
If the ContainerIPdu.headerType is set to noHeader then the ContainerIPdu.rxAcceptContainedIPdu attribute value shall be set to acceptConfigured.

[constr_xxxx3] Usage of ContainedIPduProps.updateIndicationBitPosition
ContainedIPduProps.updateIndicationBitPosition is only allowed to be set to a value if the headerType of the ContainerIPdu that contains the IPdu with containedIPduProps is set to noHeader.

[constr_xxxx4] Only the last contained IPdu (according to the ContainedIPduProps.offset) of a ContainerIPdu with static container layout (i.e., a ContainerIPdu with ContainerIPduHeaderTypeEnum set to noHeader) may be a

dynamic length PDU (i.e, a PDU that at runtime may exhibit a length different from the one statically configured via Pdu.length of the respective Pdu). All other contained PDUs of a ContainerIPdu with static container layout have to be static length PDUs.

Adapt the following spec items to:

[TPS_SYST_02098] Header id and header type of a contained IPdu
A contained IPdu shall always have the same headerId per header type (long or short header), regardless in which ContainerIPdu it is collected. If noHeader is set then the contained Ipdu does not need to have a headerId.

[TPS_SYST_02100] Relation between ContainerIPdu and contained IPdus on receiver side
On receiver side, it is not necessarily required to statically define which IPdus may be contained inside a ContainerIPdu if the header mode is used. Thus it would be possible to update the senders of ContainerIPdus and put different or additional IPdus inside.

Rework descriptions of the following classes and attributes in the model:

ContainerIPdu class description: Allows to collect several IPdus in one ContainerIPdu based on the headerType
ContainerIPdu.headerType attribute description: Defines whether and which header type is used (header id and length).

==========
Proposed solution SRS_IpduM:

- rename Chapter 6.1.4 to Dynamic I-PDU to Container Mapping
rename [SRS_IpduM_02820] to Dynamic I-PDU Mapping
add a dependency: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_0XXXX) but no mixture.
- add new Chapter Static I-PDU to Container Mapping
- add new SRS requirement: Static I-PDU Mapping
Type: Draft
Description: Multiple I-PDUs shall be mappable to a static position into a Container-PDU. If an I-PDU is mapped statically to a Container it is always transported in the same position of the Container-PDU.
Rationale: Support scenarios of Container Mapping where no header shall be used to reduce bandwidth in case always all contained PDU shall be sent.

Use Case: Efficient transmission of small PDUs on high bandwidth busses
Dependencies: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_02820) but no mixture.
–Last change on issue 76543 comment 100–

BW-C-Level:

Application	Specification	Bus
1	2	1

1.9 Specification Item ECUC_IpduM_00202

Trace References:

none

Content:

Name	IpduMContainedRxPduShortHeaderIdIpduMContainedRxPdu.IpduMContainedRxPduShort HeaderId		
Parent Container	IpduMContainedRxPdu		
Description	ShortHeader Id which is part of the ContainerPdu when this ContainedPdu is inside.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	1 .. 16777215		
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 dependency: Only valid if IpduMContainerHeaderSize is set to IPDUM_HEADERTYPE_SHORT		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76543: [IpduM] Container Pdu not usable for CAN-FD

  Problem description:

With the CAN-FD extension, the Container PDU concept has been introduced. Unfortunately this it not usable in a meaningful way for CAN-FD due to the 4-byte header for each contained PDU of a Container Pdu.

Agreed solution:

Proposed solution for the SWS IPduM:

Extend the enumeration of IpduMContainerHeaderSize by:
- IPDUM_HEADERTYPE_LONG
- IPDUM_HEADERTYPE_SHORT
- IPDUM_HEADERTYPE_NONE

The detailed proposed solution is contained here: https://svn.autosar.org/repos/work/09_WorkPackages/WP-A2/70_InternalDocuments/RfCs/76543-AUTOSAR_SWS_IPDUMultiplexer.doc

==========
Proposed solution TPS_SystemTemplate:

Extend the enumeration of ContainerIPduHeaderTypeEnum by:
- noHeader: No Header is used and the the location of each containedPdu in the ContainerPdu is statically configured.

Add the following optional attributes to ContainedIPduProps
- offset (PositiveInteger): Byte offset that describes the location of the ContainedPdu in the ContainerPdu if no header is used.
- updateIndicationBitPosition (PositiveInteger): The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated.

Add the following optional attributes to ContainerIPdu
- unusedBitPattern (PositiveInteger): IPduM fills not updated areas of the ContainerPdu with this byte-pattern.

Rewrite the Introduction of chapter 6.3.1 ContainerIPdu

For each IPdu which is put inside a ContainerIPdu, a header may be provided which determines which IPdu is contained (ContainedIPduProps.headerIdLongHeader or headerIdShortHeader and what the size of that IPdu is (DLC during runtime). With this header mode the receivers are able to extract the individual contained IPdus

again. As an alternative option to the usage of headers a statically configured layout of IPdus in the ContainerIPdu is supported.

Introduce the following constraints:

[constr_xxxx1] Mandatory offset if noHeader is used
For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu.containedPduTriggering with ContainerIPdu.headerType = noHeader the IPdu.containedIPduProps.offset shall be defined.

[constr_xxxx2] Usage of ContainerIPdu.rxAcceptContainedIPdu if noHeader is used
If the ContainerIPdu.headerType is set to noHeader then the ContainerIPdu.rxAcceptContainedIPdu attribute value shall be set to acceptConfigured.

[constr_xxxx3] Usage of ContainedIPduProps.updateIndicationBitPosition
ContainedIPduProps.updateIndicationBitPosition is only allowed to be set to a value if the headerType of the ContainerIPdu that contains the IPdu with containedIPduProps is set to noHeader.

[constr_xxxx4] Only the last contained IPdu (according to the ContainedIPduProps.offset) of a ContainerIPdu with static container layout (i.e., a ContainerIPdu with ContainerIPduHeaderTypeEnum set to noHeader) may be a dynamic length PDU (i.e, a PDU that at runtime may exhibit a length different from the one statically configured via Pdu.length of the respective Pdu). All other contained PDUs of a ContainerIPdu with static container layout have to be static length PDUs.

Adapt the following spec items to:

[TPS_SYST_02098] Header id and header type of a contained IPdu
A contained IPdu shall always have the same headerId per header type (long or short header), regardless in which ContainerIPdu it is collected. If noHeader is set then the contained Ipdu does not need to have a headerId.

[TPS_SYST_02100] Relation between ContainerIPdu and contained IPdus on receiver side
On receiver side, it is not necessarily required to statically define which IPdus may be contained inside a ContainerIPdu if the header mode is used. Thus it would be possible to update the senders of ContainerIPdus and put different or additional IPdus inside.

Rework descriptions of the following classes and attributes in the model:

ContainerIPdu class description: Allows to collect several IPdus in one ContainerIPdu based on the headerType
ContainerIPdu.headerType attribute description: Defines whether and which header type is used (header id and length).

==========
Proposed solution SRS_IpduM:

- rename Chapter 6.1.4 to Dynamic I-PDU to Container Mapping
rename [SRS_IpduM_02820] to Dynamic I-PDU Mapping
add a dependency: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_0XXXX) but no mixture.
- add new Chapter Static I-PDU to Container Mapping
- add new SRS requirement: Static I-PDU Mapping
Type: Draft
Description: Multiple I-PDUs shall be mappable to a static position into a Container-PDU. If an I-PDU is mapped statically to a Container it is always transported in the same position of the Container-PDU.
Rationale: Support scenarios of Container Mapping where no header shall be used to reduce bandwidth in case always all contained PDU shall be sent.
Use Case: Efficient transmission of small PDUs on high bandwidth busses
Dependencies: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_02820) but no mixture.
–Last change on issue 76543 comment 100–

BW-C-Level:

Application	Specification	Bus
1	2	1

1.10 Specification Item ECUC_IpduM_00203

Trace References:

none

Content:

Name	IpduMContainedRxPduLongHeaderIdIpduMContainedRxPdu.IpduMContainedRxPduLongHeaderId
Parent Container	IpduMContainedRxPdu

Description	LongHeader Id which is part of the ContainerPdu when this ContainedPdu is inside.		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	1 .. 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 dependency: Only valid if IpduMContainerHeaderSize is set to IPDUM_HEADERTYPE_LONG		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76543: [IpduM] Container Pdu not usable for CAN-FD

  Problem description:

  With the CAN-FD extension, the Container PDU concept has been introduced. Unfortunately this it not usable in a meaningful way for CAN-FD due to the 4-byte header for each contained PDU of a Container Pdu.

  Agreed solution:

  Proposed solution for the SWS IPduM:

  Extend the enumeration of IpduMContainerHeaderSize by:
  - IPDUM_HEADERTYPE_LONG
  - IPDUM_HEADERTYPE_SHORT
  - IPDUM_HEADERTYPE_NONE

  The detailed proposed solution is contained here: https://svn.autosar.org/repos/work/09_WorkPackages/WP-A2/70_InternalDocuments/RfCs/76543-AUTOSAR_SWS_IPDUMultiplexer.doc

  ==========

  Proposed solution TPS_SystemTemplate:

Extend the enumeration of ContainerIPduHeaderTypeEnum by:
- noHeader: No Header is used and the the location of each containedPdu in the ContainerPdu is statically configured.

Add the following optional attributes to ContainedIPduProps
- offset (PositiveInteger): Byte offset that describes the location of the ContainedPdu in the ContainerPdu if no header is used.
- updateIndicationBitPosition (PositiveInteger): The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated.

Add the following optional attributes to ContainerIPdu
- unusedBitPattern (PositiveInteger): IPduM fills not updated areas of the ContainerPdu with this byte-pattern.

Rewrite the Introduction of chapter 6.3.1 ContainerIPdu

For each IPdu which is put inside a ContainerIPdu, a header may be provided which determines which IPdu is contained (ContainedIPduProps.headerIdLongHeader or headerIdShortHeader and what the size of that IPdu is (DLC during runtime). With this header mode the receivers are able to extract the individual contained IPdus again. As an alternative option to the usage of headers a statically configured layout of IPdus in the ContainerIPdu is supported.

Introduce the following constraints:

[constr_xxxx1] Mandatory offset if noHeader is used
For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu.containedPduTriggering with ContainerIPdu.headerType = noHeader the IPdu.containedIPduProps.offset shall be defined.

[constr_xxxx2] Usage of ContainerIPdu.rxAcceptContainedIPdu if noHeader is used
If the ContainerIPdu.headerType is set to noHeader then the ContainerIPdu.rxAcceptContainedIPdu attribute value shall be set to acceptConfigured.

[constr_xxxx3] Usage of ContainedIPduProps.updateIndicationBitPosition
ContainedIPduProps.updateIndicationBitPosition is only allowed to be set to a value if the headerType of the ContainerIPdu that contains the IPdu with containedIPduProps is set to noHeader.

[constr_xxxx4] Only the last contained IPdu (according to the Con-

tainedIPduProps.offset) of a ContainerIPdu with static container layout (i.e., a ContainerIPdu with ContainerIPduHeaderTypeEnum set to noHeader) may be a dynamic length PDU (i.e, a PDU that at runtime may exhibit a length different from the one statically configured via Pdu.length of the respective Pdu). All other contained PDUs of a ContainerIPdu with static container layout have to be static length PDUs.

Adapt the following spec items to:

[TPS_SYST_02098] Header id and header type of a contained IPdu
A contained IPdu shall always have the same headerId per header type (long or short header), regardless in which ContainerIPdu it is collected. If noHeader is set then the contained Ipdu does not need to have a headerId.

[TPS_SYST_02100] Relation between ContainerIPdu and contained IPdus on receiver side
On receiver side, it is not necessarily required to statically define which IPdus may be contained inside a ContainerIPdu if the header mode is used. Thus it would be possible to update the senders of ContainerIPdus and put different or additional IPdus inside.

Rework descriptions of the following classes and attributes in the model:

ContainerIPdu class description: Allows to collect several IPdus in one ContainerIPdu based on the headerType
ContainerIPdu.headerType attribute description: Defines whether and which header type is used (header id and length).

==========
Proposed solution SRS_IpduM:

- rename Chapter 6.1.4 to Dynamic I-PDU to Container Mapping
rename [SRS_IpduM_02820] to Dynamic I-PDU Mapping
add a dependency: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_0XXXX) but no mixture.
- add new Chapter Static I-PDU to Container Mapping
- add new SRS requirement: Static I-PDU Mapping
Type: Draft
Description: Multiple I-PDUs shall be mappable to a static position into a Container-PDU. If an I-PDU is mapped statically to a Container it is always transported in the same position of the Container-PDU.

Rationale: Support scenarios of Container Mapping where no header shall be used to reduce bandwidth in case always all contained PDU shall be sent.
Use Case: Efficient transmission of small PDUs on high bandwidth busses
Dependencies: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_02820) but no mixture.
–Last change on issue 76543 comment 100–

BW-C-Level:

Application	Specification	Bus
1	2	1

1.11 Specification Item ECUC_IpduM_00204

Trace References:

none

Content:

Module Name	IpduM
Module Description	Configuration of the IpduM (Ipdu Multiplexer) 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
IpduMConfig	1	This container contains the sub containers of the IpduM module. * The IpduMTxPathway subcontainer includes information about sent I-PDUs. * The IpduMRxPathway includes information about received I-PDUs. * The IpduMTx ContainerMContainerTxPdu and Ipdu MTxContainedMContainedTxPdu include information about the sending of ContainerPdus. * The IpduMRx ContainerMContainerRxPdu and Ipdu MRxContainedMContainedRxPdu include information about the reception of ContainerPdus.
IpduMGeneral	1	Contains the general configuration parameters of IpduM.

Included Containers		
Container Name	Multiplicity	Scope / Dependency
IpduMPublishedInformation	1	Additional published parameters not covered by CommonPublished Information container. Note that these parameters do not have any configuration class setting, since they are published information.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73303: Misspelled container names in descriptions of configuration

  Problem description:

  The chapter ’10 Configuration specification’ provides misspelled names of containers and references in descriptions.
  This comprises the wrong names IpduMRxContainerPdu, IpduMTxContainerPdu, IpduMRxContainedPdu,IpduMTxContainedPdu and IpduMRxContainedPduContainerRef.

  Agreed solution:

  Correct all occurrences of
  IpduMRxContainerPdu by IpduMContainerRxPdu,
  IpduMTxContainerPdu by IpduMContainerTxPdu,
  IpduMRxContainedPdu by IpduMContainedRxPdu,
  IpduMTxContainedPdu by IpduMContainedTxPdu and
  IpduMRxContainedPduContainerRef by IpduMContainedRxInContainerPduRef.

  BW-C-Level:

Application	Specification	Bus
1	1	1

1.12 Specification Item ECUC_IpduM_00206

Trace References:

none

Content:

Name	IpduMContainedPduOffsetIpduMContainedTxPdu.IpduMContainedPduOffset
Parent Container	IpduMContainedTxPdu

Description	Static offset (in bytes) of the ContainedPdu. Tags: atp.Status=draft		
Multiplicity	0..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: ECU dependency: - only valid if IpduMContainerHeaderSize is set to IPDUM_HEADERTYPE_NONE. - only the ContainedPdu with the highest offset within a ContainerPdu may have variable length.		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76543: [IpduM] Container Pdu not usable for CAN-FD

  Problem description:

  With the CAN-FD extension, the Container PDU concept has been introduced. Unfortunately this it not usable in a meaningful way for CAN-FD due to the 4-byte header for each contained PDU of a Container Pdu.

  Agreed solution:

  Proposed solution for the SWS IPduM:

  Extend the enumeration of IpduMContainerHeaderSize by:
  - IPDUM_HEADERTYPE_LONG
  - IPDUM_HEADERTYPE_SHORT
  - IPDUM_HEADERTYPE_NONE

  The detailed proposed solution is contained here: https://svn.autosar.org/repos/work/09_WorkPackages/WP-A2/70_InternalDocuments/RfCs/76543-AUTOSAR_SWS_IPDUMultiplexer.doc

  ==========
  Proposed solution TPS_SystemTemplate:

Extend the enumeration of ContainerIPduHeaderTypeEnum by:
- noHeader: No Header is used and the the location of each containedPdu in the ContainerPdu is statically configured.

Add the following optional attributes to ContainedIPduProps
- offset (PositiveInteger): Byte offset that describes the location of the ContainedPdu in the ContainerPdu if no header is used.
- updateIndicationBitPosition (PositiveInteger): The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated.

Add the following optional attributes to ContainerIPdu
- unusedBitPattern (PositiveInteger): IPduM fills not updated areas of the ContainerPdu with this byte-pattern.

Rewrite the Introduction of chapter 6.3.1 ContainerIPdu

For each IPdu which is put inside a ContainerIPdu, a header may be provided which determines which IPdu is contained (ContainedIPduProps.headerIdLongHeader or headerIdShortHeader and what the size of that IPdu is (DLC during runtime). With this header mode the receivers are able to extract the individual contained IPdus again. As an alternative option to the usage of headers a statically configured layout of IPdus in the ContainerIPdu is supported.

Introduce the following constraints:

[constr_xxxx1] Mandatory offset if noHeader is used
For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu.containedPduTriggering with ContainerIPdu.headerType = noHeader the IPdu.containedIPduProps.offset shall be defined.

[constr_xxxx2] Usage of ContainerIPdu.rxAcceptContainedIPdu if noHeader is used
If the ContainerIPdu.headerType is set to noHeader then the ContainerIPdu.rxAcceptContainedIPdu attribute value shall be set to acceptConfigured.

[constr_xxxx3] Usage of ContainedIPduProps.updateIndicationBitPosition
ContainedIPduProps.updateIndicationBitPosition is only allowed to be set to a value if the headerType of the ContainerIPdu that contains the IPdu with containedIPduProps is set to noHeader.

[constr_xxxx4] Only the last contained IPdu (according to the ContainedIPduProps.offset) of a ContainerIPdu with static container layout (i.e., a ContainerIPdu with ContainerIPduHeaderTypeEnum set to noHeader) may be a dynamic length PDU (i.e, a PDU that at runtime may exhibit a length different from the one statically configured via Pdu.length of the respective Pdu). All other contained PDUs of a ContainerIPdu with static container layout have to be static length PDUs.

Adapt the following spec items to:

[TPS_SYST_02098] Header id and header type of a contained IPdu
A contained IPdu shall always have the same headerId per header type (long or short header), regardless in which ContainerIPdu it is collected. If noHeader is set then the contained Ipdu does not need to have a headerId.

[TPS_SYST_02100] Relation between ContainerIPdu and contained IPdus on receiver side
On receiver side, it is not necessarily required to statically define which IPdus may be contained inside a ContainerIPdu if the header mode is used. Thus it would be possible to update the senders of ContainerIPdus and put different or additional IPdus inside.

Rework descriptions of the following classes and attributes in the model:

ContainerIPdu class description: Allows to collect several IPdus in one ContainerIPdu based on the headerType
ContainerIPdu.headerType attribute description: Defines whether and which header type is used (header id and length).

==========
Proposed solution SRS_IpduM:

- rename Chapter 6.1.4 to Dynamic I-PDU to Container Mapping
rename [SRS_IpduM_02820] to Dynamic I-PDU Mapping
add a dependency: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_0XXXX) but no mixture.
- add new Chapter Static I-PDU to Container Mapping
- add new SRS requirement: Static I-PDU Mapping
Type: Draft
Description: Multiple I-PDUs shall be mappable to a static position into a Container-PDU. If an I-PDU is mapped statically to a Container it is always transported in the

same position of the Container-PDU.
Rationale: Support scenarios of Container Mapping where no header shall be used to reduce bandwidth in case always all contained PDU shall be sent.
Use Case: Efficient transmission of small PDUs on high bandwidth busses
Dependencies: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_02820) but no mixture.
–Last change on issue 76543 comment 100–

BW-C-Level:

Application	Specification	Bus
1	2	1

1.13 Specification Item ECUC_IpduM_00207

Trace References:

none

Content:

Name	IpduMPduUpdateBitPositionIpduMContainedTxPdu.IpduMPduUpdateBitPosition		
Parent Container	IpduMContainedTxPdu		
Description	This value specifies where the PDU's Update-Bit is stored in the Container PDU (bit location of PDU's Update-Bit in the Container PDU). Tags: atp.Status=draft		
Multiplicity	0..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 dependency: - only valid if IpduMContainerHeaderSize is set to IPDUM_HEADERTYPE_NONE.		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76543: [IpduM] Container Pdu not usable for CAN-FD

Problem description:

With the CAN-FD extension, the Container PDU concept has been introduced. Unfortunately this it not usable in a meaningful way for CAN-FD due to the 4-byte header for each contained PDU of a Container Pdu.

Agreed solution:

Proposed solution for the SWS IPduM:

Extend the enumeration of IpduMContainerHeaderSize by:
- IPDUM_HEADERTYPE_LONG
- IPDUM_HEADERTYPE_SHORT
- IPDUM_HEADERTYPE_NONE

The detailed proposed solution is contained here: https://svn.autosar.org/repos/work/09_WorkPackages/WP-A2/70_InternalDocuments/RfCs/76543-AUTOSAR_SWS_IPDUMultiplexer.doc

==========
Proposed solution TPS_SystemTemplate:

Extend the enumeration of ContainerIPduHeaderTypeEnum by:
- noHeader: No Header is used and the the location of each containedPdu in the ContainerPdu is statically configured.

Add the following optional attributes to ContainedIPduProps
- offset (PositiveInteger): Byte offset that describes the location of the ContainedPdu in the ContainerPdu if no header is used.
- updateIndicationBitPosition (PositiveInteger): The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated.

Add the following optional attributes to ContainerIPdu
- unusedBitPattern (PositiveInteger): IPduM fills not updated areas of the ContainerPdu with this byte-pattern.

Rewrite the Introduction of chapter 6.3.1 ContainerIPdu

For each IPdu which is put inside a ContainerIPdu, a header may be provided which determines which IPdu is contained (ContainedIPduProps.headerIdLongHeader or

headerIdShortHeader and what the size of that IPdu is (DLC during runtime). With this header mode the receivers are able to extract the individual contained IPdus again. As an alternative option to the usage of headers a statically configured layout of IPdus in the ContainerIPdu is supported.

Introduce the following constraints:

[constr_xxxx1] Mandatory offset if noHeader is used
For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu.containedPduTriggering with ContainerIPdu.headerType = noHeader the IPdu.containedIPduProps.offset shall be defined.

[constr_xxxx2] Usage of ContainerIPdu.rxAcceptContainedIPdu if noHeader is used
If the ContainerIPdu.headerType is set to noHeader then the ContainerIPdu.rxAcceptContainedIPdu attribute value shall be set to acceptConfigured.

[constr_xxxx3] Usage of ContainedIPduProps.updateIndicationBitPosition
ContainedIPduProps.updateIndicationBitPosition is only allowed to be set to a value if the headerType of the ContainerIPdu that contains the IPdu with containedIPduProps is set to noHeader.

[constr_xxxx4] Only the last contained IPdu (according to the ContainedIPduProps.offset) of a ContainerIPdu with static container layout (i.e., a ContainerIPdu with ContainerIPduHeaderTypeEnum set to noHeader) may be a dynamic length PDU (i.e, a PDU that at runtime may exhibit a length different from the one statically configured via Pdu.length of the respective Pdu). All other contained PDUs of a ContainerIPdu with static container layout have to be static length PDUs.

Adapt the following spec items to:

[TPS_SYST_02098] Header id and header type of a contained IPdu
A contained IPdu shall always have the same headerId per header type (long or short header), regardless in which ContainerIPdu it is collected. If noHeader is set then the contained Ipdu does not need to have a headerId.

[TPS_SYST_02100] Relation between ContainerIPdu and contained IPdus on receiver side
On receiver side, it is not necessarily required to statically define which IPdus may be contained inside a ContainerIPdu if the header mode is used. Thus it would be possible to update the senders of ContainerIPdus and put different or additional IPdus inside.

Rework descriptions of the following classes and attributes in the model:

ContainerIPdu class description: Allows to collect several IPdus in one ContainerIPdu based on the headerType
ContainerIPdu.headerType attribute description: Defines whether and which header type is used (header id and length).

==========
Proposed solution SRS_IpduM:

- rename Chapter 6.1.4 to Dynamic I-PDU to Container Mapping
rename [SRS_IpduM_02820] to Dynamic I-PDU Mapping
add a dependency: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_0XXXX) but no mixture.
- add new Chapter Static I-PDU to Container Mapping
- add new SRS requirement: Static I-PDU Mapping
Type: Draft
Description: Multiple I-PDUs shall be mappable to a static position into a Container-PDU. If an I-PDU is mapped statically to a Container it is always transported in the same position of the Container-PDU.
Rationale: Support scenarios of Container Mapping where no header shall be used to reduce bandwidth in case always all contained PDU shall be sent.
Use Case: Efficient transmission of small PDUs on high bandwidth busses
Dependencies: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_02820) but no mixture.
–Last change on issue 76543 comment 100–

BW-C-Level:

Application	Specification	Bus
1	2	1

1.14 Specification Item ECUC_IpduM_00208

Trace References:

none

Content:

Name	IpduMUnusedAreasDefaultIpduMContainerTxPdu.IpduMUnusedAreasDefault		
Parent Container	IpduMContainerTxPdu		
Description	IpduM fills not updated areas of the Container PDU with this byte-pattern. Tags: atp.Status=draft		
Multiplicity	0..1		
Type	EcucIntegerParamDef		
Range	0 .. 255		
Default value	–		
Post-Build Variant Multiplicity	false		
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: ECU dependency: Only valid if IpduMContainerHeaderSize is set to IPDUM_HEADERTYPE_NONE / should be aligned to bus-specific padding value if available.		

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76543: [IpduM] Container Pdu not usable for CAN-FD

  Problem description:

  With the CAN-FD extension, the Container PDU concept has been introduced. Unfortunately this it not usable in a meaningful way for CAN-FD due to the 4-byte header for each contained PDU of a Container Pdu.

  Agreed solution:

  Proposed solution for the SWS IPduM:

  Extend the enumeration of IpduMContainerHeaderSize by:
  - IPDUM_HEADERTYPE_LONG
  - IPDUM_HEADERTYPE_SHORT
  - IPDUM_HEADERTYPE_NONE

  The detailed proposed solution is contained here: https://svn.autosar.org/repos/work/09_WorkPackages/WP-A2/70_InternalDocuments/RfCs/76543-AUTOSAR_SWS_IPDUMultiplexer.doc

==========

Proposed solution TPS_SystemTemplate:

Extend the enumeration of ContainerIPduHeaderTypeEnum by:
- noHeader: No Header is used and the the location of each containedPdu in the ContainerPdu is statically configured.

Add the following optional attributes to ContainedIPduProps
- offset (PositiveInteger): Byte offset that describes the location of the ContainedPdu in the ContainerPdu if no header is used.
- updateIndicationBitPosition (PositiveInteger): The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated.

Add the following optional attributes to ContainerIPdu
- unusedBitPattern (PositiveInteger): IPduM fills not updated areas of the ContainerPdu with this byte-pattern.

Rewrite the Introduction of chapter 6.3.1 ContainerIPdu

For each IPdu which is put inside a ContainerIPdu, a header may be provided which determines which IPdu is contained (ContainedIPduProps.headerIdLongHeader or headerIdShortHeader and what the size of that IPdu is (DLC during runtime). With this header mode the receivers are able to extract the individual contained IPdus again. As an alternative option to the usage of headers a statically configured layout of IPdus in the ContainerIPdu is supported.

Introduce the following constraints:

[constr_xxxx1] Mandatory offset if noHeader is used
For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu.containedPduTriggering with ContainerIPdu.headerType = noHeader the IPdu.containedIPduProps.offset shall be defined.

[constr_xxxx2] Usage of ContainerIPdu.rxAcceptContainedIPdu if noHeader is used
If the ContainerIPdu.headerType is set to noHeader then the ContainerIPdu.rxAcceptContainedIPdu attribute value shall be set to acceptConfigured.

[constr_xxxx3] Usage of ContainedIPduProps.updateIndicationBitPosition
ContainedIPduProps.updateIndicationBitPosition is only allowed to be set to a

value if the headerType of the ContainerIPdu that contains the IPdu with containedIPduProps is set to noHeader.

[constr_xxxx4] Only the last contained IPdu (according to the ContainedIPduProps.offset) of a ContainerIPdu with static container layout (i.e., a ContainerIPdu with ContainerIPduHeaderTypeEnum set to noHeader) may be a dynamic length PDU (i.e, a PDU that at runtime may exhibit a length different from the one statically configured via Pdu.length of the respective Pdu). All other contained PDUs of a ContainerIPdu with static container layout have to be static length PDUs.

Adapt the following spec items to:

[TPS_SYST_02098] Header id and header type of a contained IPdu
A contained IPdu shall always have the same headerId per header type (long or short header), regardless in which ContainerIPdu it is collected. If noHeader is set then the contained Ipdu does not need to have a headerId.

[TPS_SYST_02100] Relation between ContainerIPdu and contained IPdus on receiver side
On receiver side, it is not necessarily required to statically define which IPdus may be contained inside a ContainerIPdu if the header mode is used. Thus it would be possible to update the senders of ContainerIPdus and put different or additional IPdus inside.

Rework descriptions of the following classes and attributes in the model:

ContainerIPdu class description: Allows to collect several IPdus in one ContainerIPdu based on the headerType
ContainerIPdu.headerType attribute description: Defines whether and which header type is used (header id and length).

==========
Proposed solution SRS_IpduM:

- rename Chapter 6.1.4 to Dynamic I-PDU to Container Mapping
rename [SRS_IpduM_02820] to Dynamic I-PDU Mapping
add a dependency: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_0XXXX) but no mixture.
- add new Chapter Static I-PDU to Container Mapping
- add new SRS requirement: Static I-PDU Mapping

Type: Draft
Description: Multiple I-PDUs shall be mappable to a static position into a Container-PDU. If an I-PDU is mapped statically to a Container it is always transported in the same position of the Container-PDU.
Rationale: Support scenarios of Container Mapping where no header shall be used to reduce bandwidth in case always all contained PDU shall be sent.
Use Case: Efficient transmission of small PDUs on high bandwidth busses
Dependencies: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_02820) but no mixture.
–Last change on issue 76543 comment 100–

BW-C-Level:

Application	Specification	Bus
1	2	1

1.15 Specification Item SWS_IpduM_00026

Trace References:

SRS_BSW_00337

Content:

API service called with wrong parameter:

- error code: IPDUM_E_PARAM
- value [hex]: 0x10

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77161: Tables in specification documents contain trace items

  Problem description:

  Name: Wolf-Hendrik Kaps
  Phone: 0711-811-23288
  Role: Jg-Tooling member

  Description/Motivation:

  Some specification documents contain tables which include trace items. E.g. SWS_Com, table in 7.12.1 Development Errors, SWS_Rte, Table 5.4: RTE Error and Status values.

As discussed in tooling session we should extend ValidateARXML routine to elicit trace items inside tables.

Further on we shall ensure that tracebles do not contain figures and tables (77206, 74860)
–Last change on issue 77161 comment 3–

Agreed solution:

1. Affected Documents
==========

1. Move the Traceable out of the f**ing tables (see attachment)
2. move Tables and figures out of the Traceable

2. Word2arxml and latex2arxml which is used by checkDocumentSource
==========

let these scripts complain also requested by 77206, 74860 but summarized here

* Tables in Traceable
* Traceable in Tables
* Figures in Traceable

2.1 GST: add these contraints
==========

3. CP_Tool_Scripts
==========

* let latexinstatiaator complain about Figures in Traceable
* note: tha table in traceable violates the schema and is flagged already
* let latexinstantiator complain about traceable in tables
–Last change on issue 77161 comment 7–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.16 Specification Item SWS_IpduM_00104

Trace References:

SRS_BSW_00009

Content:

API function	Description
Det_ReportRuntimeError	Service to report runtime errors. If a callout has been configured then this callout shall be called.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #59085: Rollout of 'Runtime errors'

  Problem description:

  Inconsistencies in SWS with semantics of Default errors
  –Last change on issue 59085 comment 26–

  Agreed solution:

  solution in Column "G" of the new attachment https://www.autosar.org/bugzilla/attachment.cgi?id=4604

  Notes:
  - It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
  - The review task of the ITs shall be done by the WP to which the specification "belongs".

  *** BSW UML Model ***
  SWS_CanNm:
  ———-
  Chapter 8.6.1 Optional Interfaces:
  Add within SWS_CanNm_00325 the API function Det_ReportRunTimeError

  SWS_LinIf:
  ———-
  SWS_LinIf_00359: add Det_ReportRuntimeError

  SWS_UdpNm:

———-
Replace UDPNM_E_NO_INIT with UDPNM_E_UNINIT in description of API UdpNm_MainFunction_<Instance Id> (SWS_UdpNm_00234)

*** ECUC XML ***
Not affected. No configuration of runtime error reporting required (see SWS BSW General).
–Last change on issue 59085 comment 88–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.17 Specification Item SWS_IpduM_00148

Trace References:

SRS_BSW_00415

Content:

The file IpduM.c shall include IpduM.h, IpduM_Cbk.h, PduR_IpduM.h, and Det.h, optionally IpduM_Cfg.h, Det.h and Com.h.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #59085: Rollout of 'Runtime errors'

  Problem description:

  Inconsistencies in SWS with semantics of Default errors
  –Last change on issue 59085 comment 26–

  Agreed solution:

  solution in Column "G" of the new attachment https://www.autosar.org/bugzilla/attachment.cgi?id=4604

  Notes:
  - It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
  - The review task of the ITs shall be done by the WP to which the specification "belongs".

*** BSW UML Model ***
SWS_CanNm:
———-
Chapter 8.6.1 Optional Interfaces:
Add within SWS_CanNm_00325 the API function Det_ReportRunTimeError

SWS_LinIf:
———-
SWS_LinIf_00359: add Det_ReportRuntimeError

SWS_UdpNm:
———-
Replace UDPNM_E_NO_INIT with UDPNM_E_UNINIT in description of API UdpNm_MainFunction_<Instance Id> (SWS_UdpNm_00234)

*** ECUC XML ***
Not affected. No configuration of runtime error reporting required (see SWS BSW General).
–Last change on issue 59085 comment 88–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.18 Specification Item SWS_IpduM_00153

Trace References:

SRS_BSW_00337

Content:

API service used without module initialization

- error code: IPDUM_E_ UNINIT
- value [hex]: 0x20

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77161: Tables in specification documents contain trace items

Problem description:

Name: Wolf-Hendrik Kaps
Phone: 0711-811-23288
Role: Jg-Tooling member

Description/Motivation:

Some specification documents contain tables which include trace items. E.g. SWS_Com, table in 7.12.1 Development Errors, SWS_Rte, Table 5.4: RTE Error and Status values.

As discussed in tooling session we should extend ValidateARXML routine to elicit trace items inside tables.

Further on we shall ensure that tracebles do not contain figures and tables (77206, 74860)
–Last change on issue 77161 comment 3–

Agreed solution:

1. Affected Documents
==========

1. Move the Traceable out of the f**ing tables (see attachment)
2. move Tables and figures out of the Traceable

2. Word2arxml and latex2arxml which is used by checkDocumentSource
==========

let these scripts complain also requested by 77206, 74860 but summarized here

* Tables in Traceable
* Traceable in Tables
* Figures in Traceable

2.1 GST: add these contraints
==========

3. CP_Tool_Scripts

==========

* let latexinstatiaator complain about Figures in Traceable
* note: tha table in traceable violates the schema and is flagged already
* let latexinstantiator complain about traceable in tables
–Last change on issue 77161 comment 7–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.19 Specification Item SWS_IpduM_00162

Trace References:

SRS_BSW_00337, SRS_BSW_00414

Content:

NULL pointer checking

- error code: IPDUM_E_PARAM_POINTER
- value [hex]: 0x11

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77161: Tables in specification documents contain trace items

  Problem description:

  ———————————————
  Name: Wolf-Hendrik Kaps
  Phone: 0711-811-23288
  Role: Jg-Tooling member
  ———————————————
  Description/Motivation:

  Some specification documents contain tables which include trace items. E.g. SWS_Com, table in 7.12.1 Development Errors, SWS_Rte, Table 5.4: RTE Error and Status values.

  As discussed in tooling session we should extend ValidateARXML routine to elicit trace items inside tables.

Further on we shall ensure that tracebles do not contain figures and tables (77206, 74860)
–Last change on issue 77161 comment 3–

Agreed solution:

1. Affected Documents
==========

1. Move the Traceable out of the f**ing tables (see attachment)
2. move Tables and figures out of the Traceable

2. Word2arxml and latex2arxml which is used by checkDocumentSource
==========

let these scripts complain also requested by 77206, 74860 but summarized here

* Tables in Traceable
* Traceable in Tables
* Figures in Traceable

2.1 GST: add these contraints
==========

3. CP_Tool_Scripts
==========

* let latexinstatiaator complain about Figures in Traceable
* note: tha table in traceable violates the schema and is flagged already
* let latexinstantiator complain about traceable in tables
–Last change on issue 77161 comment 7–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.20 Specification Item SWS_IpduM_00174

Trace References:

SRS_BSW_00414

Content:

Invalid configuration set selection

- error code: IPDUM_E_INIT_FAILED
- value [hex]: 0x21

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77161: Tables in specification documents contain trace items

  Problem description:

  _____

  Name: Wolf-Hendrik Kaps
  Phone: 0711-811-23288
  Role: Jg-Tooling member

  _____

  Description/Motivation:

  Some specification documents contain tables which include trace items. E.g. SWS_Com, table in 7.12.1 Development Errors, SWS_Rte, Table 5.4: RTE Error and Status values.

  As discussed in tooling session we should extend ValidateARXML routine to elicit trace items inside tables.

  Further on we shall ensure that tracebles do not contain figures and tables (77206, 74860)
  –Last change on issue 77161 comment 3–

  Agreed solution:

  1. Affected Documents
  ==========

  1. Move the Traceable out of the f**ing tables (see attachment)
  2. move Tables and figures out of the Traceable

  2. Word2arxml and latex2arxml which is used by checkDocumentSource
  ==========

let these scripts complain also requested by 77206, 74860 but summarized here

* Tables in Traceable
* Traceable in Tables
* Figures in Traceable

2.1 GST: add these contraints
==========

3. CP_Tool_Scripts
==========

* let latexinstatiaator complain about Figures in Traceable
* note: tha table in traceable violates the schema and is flagged already
* let latexinstantiator complain about traceable in tables
–Last change on issue 77161 comment 7–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.21 Specification Item SWS_IpduM_00175

Trace References:

SRS_IpduM_02820

Content:

Inside a dynamic Container PDU IpduM shall place the header of a contained I-PDU in front of the contained I-PDU.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76543: [IpduM] Container Pdu not usable for CAN-FD

  Problem description:

  With the CAN-FD extension, the Container PDU concept has been introduced. Unfortunately this it not usable in a meaningful way for CAN-FD due to the 4-byte header for each contained PDU of a Container Pdu.

  Agreed solution:

Proposed solution for the SWS IPduM:

Extend the enumeration of IpduMContainerHeaderSize by:
- IPDUM_HEADERTYPE_LONG
- IPDUM_HEADERTYPE_SHORT
- IPDUM_HEADERTYPE_NONE

The detailed proposed solution is contained here: https://svn.autosar.org/repos/work/09_WorkPackages/WP-A2/70_InternalDocuments/RfCs/76543-AUTOSAR_SWS_IPDUMultiplexer.doc

==========
Proposed solution TPS_SystemTemplate:

Extend the enumeration of ContainerIPduHeaderTypeEnum by:
- noHeader: No Header is used and the the location of each containedPdu in the ContainerPdu is statically configured.

Add the following optional attributes to ContainedIPduProps
- offset (PositiveInteger): Byte offset that describes the location of the ContainedPdu in the ContainerPdu if no header is used.
- updateIndicationBitPosition (PositiveInteger): The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated.

Add the following optional attributes to ContainerIPdu
- unusedBitPattern (PositiveInteger): IPduM fills not updated areas of the ContainerPdu with this byte-pattern.

Rewrite the Introduction of chapter 6.3.1 ContainerIPdu

For each IPdu which is put inside a ContainerIPdu, a header may be provided which determines which IPdu is contained (ContainedIPduProps.headerIdLongHeader or headerIdShortHeader and what the size of that IPdu is (DLC during runtime). With this header mode the receivers are able to extract the individual contained IPdus again. As an alternative option to the usage of headers a statically configured layout of IPdus in the ContainerIPdu is supported.

Introduce the following constraints:

[constr_xxxx1] Mandatory offset if noHeader is used
For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu.containedPduTriggering with ContainerIPdu.headerType = noHeader the IPdu.containedIPduProps.offset shall be defined.

[constr_xxxx2] Usage of ContainerIPdu.rxAcceptContainedIPdu if noHeader is used
If the ContainerIPdu.headerType is set to noHeader then the ContainerIPdu.rxAcceptContainedIPdu attribute value shall be set to acceptConfigured.

[constr_xxxx3] Usage of ContainedIPduProps.updateIndicationBitPosition
ContainedIPduProps.updateIndicationBitPosition is only allowed to be set to a value if the headerType of the ContainerIPdu that contains the IPdu with containedIPduProps is set to noHeader.

[constr_xxxx4] Only the last contained IPdu (according to the ContainedIPduProps.offset) of a ContainerIPdu with static container layout (i.e., a ContainerIPdu with ContainerIPduHeaderTypeEnum set to noHeader) may be a dynamic length PDU (i.e, a PDU that at runtime may exhibit a length different from the one statically configured via Pdu.length of the respective Pdu). All other contained PDUs of a ContainerIPdu with static container layout have to be static length PDUs.

Adapt the following spec items to:

[TPS_SYST_02098] Header id and header type of a contained IPdu
A contained IPdu shall always have the same headerId per header type (long or short header), regardless in which ContainerIPdu it is collected. If noHeader is set then the contained Ipdu does not need to have a headerId.

[TPS_SYST_02100] Relation between ContainerIPdu and contained IPdus on receiver side
On receiver side, it is not necessarily required to statically define which IPdus may be contained inside a ContainerIPdu if the header mode is used. Thus it would be possible to update the senders of ContainerIPdus and put different or additional IPdus inside.

Rework descriptions of the following classes and attributes in the model:

ContainerIPdu class description: Allows to collect several IPdus in one ContainerIPdu based on the headerType
ContainerIPdu.headerType attribute description: Defines whether and which header

type is used (header id and length).

==========
Proposed solution SRS_IpduM:

- rename Chapter 6.1.4 to Dynamic I-PDU to Container Mapping
rename [SRS_IpduM_02820] to Dynamic I-PDU Mapping
add a dependency: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_0XXXX) but no mixture.
- add new Chapter Static I-PDU to Container Mapping
- add new SRS requirement: Static I-PDU Mapping
Type: Draft
Description: Multiple I-PDUs shall be mappable to a static position into a Container-PDU. If an I-PDU is mapped statically to a Container it is always transported in the same position of the Container-PDU.
Rationale: Support scenarios of Container Mapping where no header shall be used to reduce bandwidth in case always all contained PDU shall be sent.
Use Case: Efficient transmission of small PDUs on high bandwidth busses
Dependencies: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_02820) but no mixture.
–Last change on issue 76543 comment 100–

BW-C-Level:

Application	Specification	Bus
1	2	1

1.22 Specification Item SWS_IpduM_00178

Trace References:

SRS_IpduM_02823

Content:

Placing of headers and payloads of contained I-PDUs inside a dynamic Container PDU shall be contiguous without any gap.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76543: [IpduM] Container Pdu not usable for CAN-FD

  Problem description:

With the CAN-FD extension, the Container PDU concept has been introduced. Unfortunately this it not usable in a meaningful way for CAN-FD due to the 4-byte header for each contained PDU of a Container Pdu.

Agreed solution:

Proposed solution for the SWS IPduM:

Extend the enumeration of IpduMContainerHeaderSize by:
- IPDUM_HEADERTYPE_LONG
- IPDUM_HEADERTYPE_SHORT
- IPDUM_HEADERTYPE_NONE

The detailed proposed solution is contained here: https://svn.autosar.org/repos/work/09_WorkPackages/WP-A2/70_InternalDocuments/RfCs/76543-AUTOSAR_SWS_IPDUMultiplexer.doc

==========
Proposed solution TPS_SystemTemplate:

Extend the enumeration of ContainerIPduHeaderTypeEnum by:
- noHeader: No Header is used and the the location of each containedPdu in the ContainerPdu is statically configured.

Add the following optional attributes to ContainedIPduProps
- offset (PositiveInteger): Byte offset that describes the location of the ContainedPdu in the ContainerPdu if no header is used.
- updateIndicationBitPosition (PositiveInteger): The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated.

Add the following optional attributes to ContainerIPdu
- unusedBitPattern (PositiveInteger): IPduM fills not updated areas of the ContainerPdu with this byte-pattern.

Rewrite the Introduction of chapter 6.3.1 ContainerIPdu

For each IPdu which is put inside a ContainerIPdu, a header may be provided which determines which IPdu is contained (ContainedIPduProps.headerIdLongHeader or headerIdShortHeader and what the size of that IPdu is (DLC during runtime). With this header mode the receivers are able to extract the individual contained IPdus

again. As an alternative option to the usage of headers a statically configured layout of IPdus in the ContainerIPdu is supported.

Introduce the following constraints:

[constr_xxxx1] Mandatory offset if noHeader is used
For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu.containedPduTriggering with ContainerIPdu.headerType = noHeader the IPdu.containedIPduProps.offset shall be defined.

[constr_xxxx2] Usage of ContainerIPdu.rxAcceptContainedIPdu if noHeader is used
If the ContainerIPdu.headerType is set to noHeader then the ContainerIPdu.rxAcceptContainedIPdu attribute value shall be set to acceptConfigured.

[constr_xxxx3] Usage of ContainedIPduProps.updateIndicationBitPosition
ContainedIPduProps.updateIndicationBitPosition is only allowed to be set to a value if the headerType of the ContainerIPdu that contains the IPdu with containedIPduProps is set to noHeader.

[constr_xxxx4] Only the last contained IPdu (according to the ContainedIPduProps.offset) of a ContainerIPdu with static container layout (i.e., a ContainerIPdu with ContainerIPduHeaderTypeEnum set to noHeader) may be a dynamic length PDU (i.e, a PDU that at runtime may exhibit a length different from the one statically configured via Pdu.length of the respective Pdu). All other contained PDUs of a ContainerIPdu with static container layout have to be static length PDUs.

Adapt the following spec items to:

[TPS_SYST_02098] Header id and header type of a contained IPdu
A contained IPdu shall always have the same headerId per header type (long or short header), regardless in which ContainerIPdu it is collected. If noHeader is set then the contained Ipdu does not need to have a headerId.

[TPS_SYST_02100] Relation between ContainerIPdu and contained IPdus on receiver side
On receiver side, it is not necessarily required to statically define which IPdus may be contained inside a ContainerIPdu if the header mode is used. Thus it would be possible to update the senders of ContainerIPdus and put different or additional IPdus inside.

Rework descriptions of the following classes and attributes in the model:

ContainerIPdu class description: Allows to collect several IPdus in one ContainerIPdu based on the headerType
ContainerIPdu.headerType attribute description: Defines whether and which header type is used (header id and length).

==========
Proposed solution SRS_IpduM:

- rename Chapter 6.1.4 to Dynamic I-PDU to Container Mapping
rename [SRS_IpduM_02820] to Dynamic I-PDU Mapping
add a dependency: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_0XXXX) but no mixture.
- add new Chapter Static I-PDU to Container Mapping
- add new SRS requirement: Static I-PDU Mapping
Type: Draft
Description: Multiple I-PDUs shall be mappable to a static position into a Container-PDU. If an I-PDU is mapped statically to a Container it is always transported in the same position of the Container-PDU.
Rationale: Support scenarios of Container Mapping where no header shall be used to reduce bandwidth in case always all contained PDU shall be sent.
Use Case: Efficient transmission of small PDUs on high bandwidth busses
Dependencies: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_02820) but no mixture.
–Last change on issue 76543 comment 100–

BW-C-Level:

Application	Specification	Bus
1	2	1

1.23 Specification Item SWS_IpduM_00184

Trace References:

SRS_IpduM_02820

Content:

When adding the first contained I-PDU to a Container PDU , and either IpduMContainer TxSendTimeout of the Container PDU or IpduMContainedTxPduSendTimeout of the contained I-PDU is configured greater than zero, the IpduM module shall start the transmis-

sion timer of the Container PDU. The timer shall be initialized with either the Container PDU's timeout (the smaller non zero value of IpduMContainerTxSendTimeout) or the contained I-PDU's timeout (and IpduMContainedTxPduSendTimeout) whichever is smaller.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77175: [IpduM] Containers without timeout

  Problem description:

  Description/Motivation:
  Currently IpduM strictly requires an assignment of a timeout to a Container or all its contained I-PDUs (SWS_IpduM_00218) whereas in the System Template it is allowed to skip the timeouts.
  Since there are use cases where this is legitimate (one of them is Containers having IpduMContainerTxFirstContainedPduTrigger=TRUE) SWS_IpduM_00218 shall be dropped. The text below SWS_IpduM_00184 should be relocated below SWS_IpduM_00201.

  [SWS_IpduM_00218] If the IpduMContainerTxSendTimeout is omitted all IpduMContainedTxPdu have to provide a IpduMContainedTxPduSendTimeout.

  [SWS_IpduM_00184] When adding the first contained I-PDU to a Container PDU, IpduM shall start the transmission timer of the Container PDU. The timer shall be initialized with either the Container PDUs timeout (IpduMContainerTxSendTimeout) or the contained I-PDUs timeout (IpduMContainedTxPduSendTimeout) whichever is smaller. (SRS_IpduM_02820)

  Rationale: This way a transmission is requested for a time-triggered bus.

  Until the Container PDU is fetched (see SWS_IpduM_00194) or unless maximum size of the Container PDU is not exceeded further requested I-PDUs assigned to this container can be added.

  Was there already a decision?

  Agreed solution:

  1) drop SWS_IpduM_00218 (remove restriction)

  2) remove "Rationale: This way a transmission is requested for a time-triggered

bus."
below SWS_IpduM_00184 (actually the rationale already exists also at the correct place)

3) update SWS_IpduM_00184 to:
SWS_IpduM_00184: When adding the first contained I-PDU to a Container PDU and either IpduMContainerTxSendTimeout of the Container PDU or IpduMContainedTxPduSendTimeout of the contained I-PDU is configured greater than zero, the IpduM module shall start the transmission timer of the Container PDU. The timer shall be initialized with smaller non zero value of IpduMContainerTxSendTimeout and IpduMContainedTxPduSendTimeout.
–Last change on issue 77175 comment 2–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.24 Specification Item SWS_IpduM_00199

Trace References:

SRS_IpduM_02820

Content:

If creating a new instance of a Container PDU would exceed IpduMContainerQueueSize the oldest instance shall be discarded. If IpduMContainerQueueSize is not configured the local instance shall be discarded.

If Development Error Detection is configured (ECUC_IpduM_00132) In both cases IPDUM_E_QUEUEOVFL shall be reported to DET via Det_ReportRuntimeError.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #59085: Rollout of 'Runtime errors'

  Problem description:

  Inconsistencies in SWS with semantics of Default errors
  –Last change on issue 59085 comment 26–

  Agreed solution:

  solution in Column "G" of the new attachment https://www.autosar.org/bugzilla/attachment.cgi?id=4604

Notes:
- It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
- The review task of the ITs shall be done by the WP to which the specification "belongs".

*** BSW UML Model ***
SWS_CanNm:
———-
Chapter 8.6.1 Optional Interfaces:
Add within SWS_CanNm_00325 the API function Det_ReportRunTimeError

SWS_LinIf:
———-
SWS_LinIf_00359: add Det_ReportRuntimeError

SWS_UdpNm:
———-
Replace UDPNM_E_NO_INIT with UDPNM_E_UNINIT in description of API UdpNm_MainFunction_<Instance Id> (SWS_UdpNm_00234)

*** ECUC XML ***
Not affected. No configuration of runtime error reporting required (see SWS BSW General).
–Last change on issue 59085 comment 88–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.25 Specification Item SWS_IpduM_00212

Trace References:

SRS_IpduM_02820

Content:

If receiving a new instance of a Container PDU would exceed IpduMContainerQueue Size the oldest instance shall be discarded . If Development Error Detection is configured (ECUC_IpduM_00132) and IPDUM_E_QUEUEOVFL shall be reported to DET via Det_ReportRuntimeError.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #59085: Rollout of 'Runtime errors'

  Problem description:

  Inconsistencies in SWS with semantics of Default errors
  –Last change on issue 59085 comment 26–

  Agreed solution:

  solution in Column "G" of the new attachment https://www.autosar.org/bugzilla/attachment.cgi?id=4604

  Notes:
  - It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
  - The review task of the ITs shall be done by the WP to which the specification "belongs".

  *** BSW UML Model ***
  SWS_CanNm:
  ———-
  Chapter 8.6.1 Optional Interfaces:
  Add within SWS_CanNm_00325 the API function Det_ReportRunTimeError

  SWS_LinIf:
  ———-
  SWS_LinIf_00359: add Det_ReportRuntimeError

  SWS_UdpNm:
  ———-
  Replace UDPNM_E_NO_INIT with UDPNM_E_UNINIT in description of API UdpNm_MainFunction_<Instance Id> (SWS_UdpNm_00234)

  *** ECUC XML ***
  Not affected. No configuration of runtime error reporting required (see SWS BSW

General).
–Last change on issue 59085 comment 88–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.26 Specification Item SWS_IpduM_00213

Trace References:

SRS_IpduM_02820

Content:

When processing a received Container PDU and detecting a header where the payload length exceeds the remaining bytes of the container the processing for this Container PDU shall be stopped and the remaining bytes shall be ignored. If Development Error Detection is configured (ECUC_IpduM_00132) Furthermore, IPDUM_E_HEADER shall be reported to DET via Det_ReportRuntimeError.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #59085: Rollout of 'Runtime errors'

  Problem description:

  Inconsistencies in SWS with semantics of Default errors
  –Last change on issue 59085 comment 26–

  Agreed solution:

  solution in Column "G" of the new attachment https://www.autosar.org/bugzilla/attachment.cgi?id=4604

  Notes:
  - It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
  - The review task of the ITs shall be done by the WP to which the specification "belongs".

  *** BSW UML Model ***
  SWS_CanNm:

———-
Chapter 8.6.1 Optional Interfaces:
Add within SWS_CanNm_00325 the API function Det_ReportRunTimeError

SWS_LinIf:
———-
SWS_LinIf_00359: add Det_ReportRuntimeError

SWS_UdpNm:
———-
Replace UDPNM_E_NO_INIT with UDPNM_E_UNINIT in description of API UdpNm_MainFunction_<Instance Id> (SWS_UdpNm_00234)

*** ECUC XML ***
Not affected. No configuration of runtime error reporting required (see SWS BSW General).
–Last change on issue 59085 comment 88–

BW-C-Level:

Application	Specification	Bus
1	4	1

1.27 Specification Item SWS_IpduM_00215

Trace References:

SRS_IpduM_02820

Content:

Erroneous header detected .

- error code: IPDUM_E_HEADER
- value [hex]: 0x30

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #59085: Rollout of ’Runtime errors’

  Problem description:

  Inconsistencies in SWS with semantics of Default errors
  –Last change on issue 59085 comment 26–

Agreed solution:

solution in Column "G" of the new attachment https://www.autosar.org/bugzilla/attachment.cgi?id=4604

Notes:
- It is not enough just to migrate the error from one classification table to another. Please also check the related requirements (and background information) which is referring to that error and adapt them if needed.
- The review task of the ITs shall be done by the WP to which the specification "belongs".

*** BSW UML Model ***
SWS_CanNm:
———-
Chapter 8.6.1 Optional Interfaces:
Add within SWS_CanNm_00325 the API function Det_ReportRunTimeError

SWS_LinIf:
———-
SWS_LinIf_00359: add Det_ReportRuntimeError

SWS_UdpNm:
———-
Replace UDPNM_E_NO_INIT with UDPNM_E_UNINIT in description of API UdpNm_MainFunction_<Instance Id> (SWS_UdpNm_00234)

*** ECUC XML ***
Not affected. No configuration of runtime error reporting required (see SWS BSW General).
–Last change on issue 59085 comment 88–

BW-C-Level:

Application	Specification	Bus
1	4	1

- RfC #77161: Tables in specification documents contain trace items

Problem description:

———————————

Name: Wolf-Hendrik Kaps
Phone: 0711-811-23288

Role: Jg-Tooling member

Description/Motivation:

Some specification documents contain tables which include trace items. E.g. SWS_Com, table in 7.12.1 Development Errors, SWS_Rte, Table 5.4: RTE Error and Status values.

As discussed in tooling session we should extend ValidateARXML routine to elicit trace items inside tables.

Further on we shall ensure that tracebles do not contain figures and tables (77206, 74860)
–Last change on issue 77161 comment 3–

Agreed solution:

1. Affected Documents
==========

1. Move the Traceable out of the f**ing tables (see attachment)
2. move Tables and figures out of the Traceable

2. Word2arxml and latex2arxml which is used by checkDocumentSource
==========

let these scripts complain also requested by 77206, 74860 but summarized here

* Tables in Traceable
* Traceable in Tables
* Figures in Traceable

2.1 GST: add these contraints
==========

3. CP_Tool_Scripts
==========

* let latexinstatiaator complain about Figures in Traceable
* note: tha table in traceable violates the schema and is flagged already

* let latexinstantiator complain about traceable in tables
–Last change on issue 77161 comment 7–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.28 Specification Item SWS_IpduM_00216

Trace References:

SRS_IpduM_02820

Content:

Container Queue overflow

- error code: IPDUM_E_QUEUEOVFL
- value [hex]: 0x31

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77161: Tables in specification documents contain trace items

  Problem description:

  ———————————
  Name: Wolf-Hendrik Kaps
  Phone: 0711-811-23288
  Role: Jg-Tooling member
  ———————————
  Description/Motivation:

  Some specification documents contain tables which include trace items. E.g. SWS_Com, table in 7.12.1 Development Errors, SWS_Rte, Table 5.4: RTE Error and Status values.

  As discussed in tooling session we should extend ValidateARXML routine to elicit trace items inside tables.

  Further on we shall ensure that tracebles do not contain figures and tables (77206, 74860)
  –Last change on issue 77161 comment 3–

Agreed solution:

1. Affected Documents
==========

1. Move the Traceable out of the f**ing tables (see attachment)
2. move Tables and figures out of the Traceable

2. Word2arxml and latex2arxml which is used by checkDocumentSource
==========

let these scripts complain also requested by 77206, 74860 but summarized here

* Tables in Traceable
* Traceable in Tables
* Figures in Traceable

2.1 GST: add these contraints
==========

3. CP_Tool_Scripts
==========

* let latexinstatiaator complain about Figures in Traceable
* note: tha table in traceable violates the schema and is flagged already
* let latexinstantiator complain about traceable in tables
–Last change on issue 77161 comment 7–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.29 Specification Item SWS_IpduM_00218

Trace References:

SRS_IpduM_02820

Content:

If the IpduMContainerTxSendTimeout is omitted all IpduMContainedTxPdu have to provide a IpduMContainedTxPduSendTimeout.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77175: [IpduM] Containers without timeout

  Problem description:

  Description/Motivation:
  Currently IpduM strictly requires an assignment of a timeout to a Container or all its contained I-PDUs (SWS_IpduM_00218) whereas in the System Template it is allowed to skip the timeouts.
  Since there are use cases where this is legitimate (one of them is Containers having IpduMContainerTxFirstContainedPduTrigger=TRUE) SWS_IpduM_00218 shall be dropped. The text below SWS_IpduM_00184 should be relocated below SWS_IpduM_00201.

  [SWS_IpduM_00218] If the IpduMContainerTxSendTimeout is omitted all IpduMContainedTxPdu have to provide a IpduMContainedTxPduSendTimeout.

  [SWS_IpduM_00184] When adding the first contained I-PDU to a Container PDU, IpduM shall start the transmission timer of the Container PDU. The timer shall be initialized with either the Container PDUs timeout (IpduMContainerTxSendTimeout) or the contained I-PDUs timeout (IpduMContainedTxPduSendTimeout) whichever is smaller. (SRS_IpduM_02820)

  Rationale: This way a transmission is requested for a time-triggered bus.

  Until the Container PDU is fetched (see SWS_IpduM_00194) or unless maximum size of the Container PDU is not exceeded further requested I-PDUs assigned to this container can be added.

  Was there already a decision?

  Agreed solution:

  1) drop SWS_IpduM_00218 (remove restriction)

  2) remove "Rationale: This way a transmission is requested for a time-triggered bus."

below SWS_IpduM_00184 (actually the rationale already exists also at the correct place)

3) update SWS_IpduM_00184 to:
SWS_IpduM_00184: When adding the first contained I-PDU to a Container PDU and either IpduMContainerTxSendTimeout of the Container PDU or IpduMContainedTxPduSendTimeout of the contained I-PDU is configured greater than zero, the IpduM module shall start the transmission timer of the Container PDU. The timer shall be initialized with smaller non zero value of IpduMContainerTxSendTimeout and IpduMContainedTxPduSendTimeout.
–Last change on issue 77175 comment 2–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.30 Specification Item SWS_IpduM_00221

Trace References:

SRS_IpduM_02821

Content:

When storing contained I-PDUs into Container PDUs, the IpduM shall store contained I-PDUs in the Container PDU in the same retain the order in which they the contained I-PDUs are passed to IpduM. If the Container PDU already contains an instance of That is the first passed contained I-PDU is placed at the beginning at the container and so on. If a contained I-PDU with IpduMContainedTxPduCollectionSemantics set to IPDUM_COLLECT_LAST_IS_BEST , is passed multiple times, the IpduM shall replace the already existing instance without modifying the order of the contained I-PDUs already collectedstore it only once at the position matching its first occurrence.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77380: [IpduM] I-PDU transmission request shall be stored in case of IpduMContainedTxPduCollectionSemantics set to IPDUM_COLLECT_LAST_IS_BEST

  Problem description:

  In ch. "Triggered Transmission and Last-is-Best semantics"
  [SWS_IpduM_00221] states:
  "
  IpduM shall store contained I-PDUs in the Container PDU

...
"

That's correct in the context of the transmission call (either IpduM_TriggerTransmit or IpduM_Transmit), when the PDUs are fetched from the upper layer.
But not for the case when the upper layer request to consider a certain I-PDU in a ContainerIPdu. In this case only the request to consider a certain I-PDU should be stored.
–Last change on issue 77380 comment 2–

Agreed solution:

[SWS_IpduM_00221] When storing contained I-PDUs into Container PDUs, the IpduM shall retain the order in which the contained I-PDUs are passed to IpduM. That is the first passed contained I-PDU is placed at the beginning at the container and so on. If a contained I-PDU with IpduMContainedTxPduCollectionSemantics set to IPDUM_COLLECT_LAST_IS_BEST is passed multiple times, the IpduM shall store it only once at the position matching its first occurrence. (SRS_IpduM_02821)

Note: The requirement above only defines the order of the contained I-PDUs but not the point in time when the container is constructed. For container with contained I-PDUs having last-is-best semantic, it might be more efficient to just keep track of the order and construct the Container PDU on demand.
–Last change on issue 77380 comment 6–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.31 Specification Item SWS_IpduM_00231

Trace References:

SRS_IpduM_02820

Content:

If in case of updating contained I-PDUs with IpduMContainedTxPduCollectionSemantics IPDUM_COLLECT_LAST_IS_BEST and the container size is not sufficient for a contained I-PDU, this contained I-PDU and all following shall be shifted to the beginning of the next container instance.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #75373: [IPduM] IPDUM_COLLECT_LAST_IS_BEST does not work with contained PDUs with dynamic size

  Problem description:

  If IPDUM_COLLECT_LAST_IS_BEST is used for a contained PDUs that increases its size after it was registered for a container PDU, the container might overflow.
  –Last change on issue 75373 comment 4–

  Agreed solution:

  Add notes and new requirement below SWS_IpduM_00220:

  In case contained I-PDUs with IpduMContainedTxPduCollectionSemantics set to IPDUM_COLLECT_LAST_IS_BEST, the IpduM module updates these I-PDUs before sending. If such contained I-PDUs have dynamic size, it can happen that the container size is not sufficient for all contained I-PDUs, if the overall size of the updated I-PDUs increases.

  SWS_IpduM_XXX1: If in case of updating contained I-PDUs with IpduMContainedTxPduCollectionSemantics IPDUM_COLLECT_LAST_IS_BEST and the container size is not sufficient for a contained I-PDU, this contained I-PDU and all following shall be shifted to the beginning of the next container instance.

  In order to preserve the order of the contained I-PDUs, also all following contained I-PDUs needs to be shifted even if there would be enough space in the current container.
  –Last change on issue 75373 comment 7–

  BW-C-Level:

Application	Specification	Bus
1	1	1

1.32 Specification Item SWS_IpduM_00232

Trace References:

SRS_IpduM_02825

Content:

If the IpduMContainerHeaderSize is set to IPDUM_HEADERTYPE_NONE, the IpduM module shall statically place the contained I-PDUs within the Container PDU according to their configured IpduMContainedTxPduOffset

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76543: [IpduM] Container Pdu not usable for CAN-FD

  Problem description:

  With the CAN-FD extension, the Container PDU concept has been introduced. Unfortunately this it not usable in a meaningful way for CAN-FD due to the 4-byte header for each contained PDU of a Container Pdu.

  Agreed solution:

  Proposed solution for the SWS IPduM:

  Extend the enumeration of IpduMContainerHeaderSize by:
  - IPDUM_HEADERTYPE_LONG
  - IPDUM_HEADERTYPE_SHORT
  - IPDUM_HEADERTYPE_NONE

  The detailed proposed solution is contained here: https://svn.autosar.org/repos/work/09_WorkPackages/WP-A2/70_InternalDocuments/RfCs/76543-AUTOSAR_SWS_IPDUMultiplexer.doc

  ==========
  Proposed solution TPS_SystemTemplate:

  Extend the enumeration of ContainerIPduHeaderTypeEnum by:
  - noHeader: No Header is used and the the location of each containedPdu in the ContainerPdu is statically configured.

  Add the following optional attributes to ContainedIPduProps
  - offset (PositiveInteger): Byte offset that describes the location of the ContainedPdu in the ContainerPdu if no header is used.
  - updateIndicationBitPosition (PositiveInteger): The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated.

  Add the following optional attributes to ContainerIPdu
  - unusedBitPattern (PositiveInteger): IPduM fills not updated areas of the ContainerPdu with this byte-pattern.

  Rewrite the Introduction of chapter 6.3.1 ContainerIPdu

For each IPdu which is put inside a ContainerIPdu, a header may be provided which determines which IPdu is contained (ContainedIPduProps.headerIdLongHeader or headerIdShortHeader and what the size of that IPdu is (DLC during runtime). With this header mode the receivers are able to extract the individual contained IPdus again. As an alternative option to the usage of headers a statically configured layout of IPdus in the ContainerIPdu is supported.

Introduce the following constraints:

[constr_xxxx1] Mandatory offset if noHeader is used
For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu.containedPduTriggering with ContainerIPdu.headerType = noHeader the IPdu.containedIPduProps.offset shall be defined.

[constr_xxxx2] Usage of ContainerIPdu.rxAcceptContainedIPdu if noHeader is used
If the ContainerIPdu.headerType is set to noHeader then the ContainerIPdu.rxAcceptContainedIPdu attribute value shall be set to acceptConfigured.

[constr_xxxx3] Usage of ContainedIPduProps.updateIndicationBitPosition
ContainedIPduProps.updateIndicationBitPosition is only allowed to be set to a value if the headerType of the ContainerIPdu that contains the IPdu with containedIPduProps is set to noHeader.

[constr_xxxx4] Only the last contained IPdu (according to the ContainedIPduProps.offset) of a ContainerIPdu with static container layout (i.e., a ContainerIPdu with ContainerIPduHeaderTypeEnum set to noHeader) may be a dynamic length PDU (i.e, a PDU that at runtime may exhibit a length different from the one statically configured via Pdu.length of the respective Pdu). All other contained PDUs of a ContainerIPdu with static container layout have to be static length PDUs.

Adapt the following spec items to:

[TPS_SYST_02098] Header id and header type of a contained IPdu
A contained IPdu shall always have the same headerId per header type (long or short header), regardless in which ContainerIPdu it is collected. If noHeader is set then the contained Ipdu does not need to have a headerId.

[TPS_SYST_02100] Relation between ContainerIPdu and contained IPdus on receiver side
On receiver side, it is not necessarily required to statically define which IPdus may

be contained inside a ContainerIPdu if the header mode is used. Thus it would be possible to update the senders of ContainerIPdus and put different or additional IPdus inside.

Rework descriptions of the following classes and attributes in the model:

ContainerIPdu class description: Allows to collect several IPdus in one ContainerIPdu based on the headerType
ContainerIPdu.headerType attribute description: Defines whether and which header type is used (header id and length).

==========
Proposed solution SRS_IpduM:

- rename Chapter 6.1.4 to Dynamic I-PDU to Container Mapping
rename [SRS_IpduM_02820] to Dynamic I-PDU Mapping
add a dependency: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_0XXXX) but no mixture.
- add new Chapter Static I-PDU to Container Mapping
- add new SRS requirement: Static I-PDU Mapping
Type: Draft
Description: Multiple I-PDUs shall be mappable to a static position into a Container-PDU. If an I-PDU is mapped statically to a Container it is always transported in the same position of the Container-PDU.
Rationale: Support scenarios of Container Mapping where no header shall be used to reduce bandwidth in case always all contained PDU shall be sent.
Use Case: Efficient transmission of small PDUs on high bandwidth busses
Dependencies: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_02820) but no mixture.
–Last change on issue 76543 comment 100–

BW-C-Level:

Application	Specification	Bus
1	2	1

1.33 Specification Item SWS_IpduM_00233

Trace References:

SRS_IpduM_02825

Content:

In case a Static Container has a configured IpduMUnusedAreasDefault, the IpduM shall ensure that all not updated areas of the Container are set to the value of IpduMUnused AreasDefault before the Container PDU is sent.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76543: [IpduM] Container Pdu not usable for CAN-FD

  Problem description:

  With the CAN-FD extension, the Container PDU concept has been introduced. Unfortunately this it not usable in a meaningful way for CAN-FD due to the 4-byte header for each contained PDU of a Container Pdu.

  Agreed solution:

  Proposed solution for the SWS IPduM:

  Extend the enumeration of IpduMContainerHeaderSize by:
  - IPDUM_HEADERTYPE_LONG
  - IPDUM_HEADERTYPE_SHORT
  - IPDUM_HEADERTYPE_NONE

  The detailed proposed solution is contained here: https://svn.autosar.org/repos/work/09_WorkPackages/WP-A2/70_InternalDocuments/RfCs/76543-AUTOSAR_SWS_IPDUMultiplexer.doc

  ==========
  Proposed solution TPS_SystemTemplate:

  Extend the enumeration of ContainerIPduHeaderTypeEnum by:
  - noHeader: No Header is used and the the location of each containedPdu in the ContainerPdu is statically configured.

  Add the following optional attributes to ContainedIPduProps
  - offset (PositiveInteger): Byte offset that describes the location of the ContainedPdu in the ContainerPdu if no header is used.
  - updateIndicationBitPosition (PositiveInteger): The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated.

Add the following optional attributes to ContainerIPdu
- unusedBitPattern (PositiveInteger): IPduM fills not updated areas of the ContainerPdu with this byte-pattern.

Rewrite the Introduction of chapter 6.3.1 ContainerIPdu

For each IPdu which is put inside a ContainerIPdu, a header may be provided which determines which IPdu is contained (ContainedIPduProps.headerIdLongHeader or headerIdShortHeader and what the size of that IPdu is (DLC during runtime). With this header mode the receivers are able to extract the individual contained IPdus again. As an alternative option to the usage of headers a statically configured layout of IPdus in the ContainerIPdu is supported.

Introduce the following constraints:

[constr_xxxx1] Mandatory offset if noHeader is used
For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu.containedPduTriggering with ContainerIPdu.headerType = noHeader the IPdu.containedIPduProps.offset shall be defined.

[constr_xxxx2] Usage of ContainerIPdu.rxAcceptContainedIPdu if noHeader is used
If the ContainerIPdu.headerType is set to noHeader then the ContainerIPdu.rxAcceptContainedIPdu attribute value shall be set to acceptConfigured.

[constr_xxxx3] Usage of ContainedIPduProps.updateIndicationBitPosition
ContainedIPduProps.updateIndicationBitPosition is only allowed to be set to a value if the headerType of the ContainerIPdu that contains the IPdu with containedIPduProps is set to noHeader.

[constr_xxxx4] Only the last contained IPdu (according to the ContainedIPduProps.offset) of a ContainerIPdu with static container layout (i.e., a ContainerIPdu with ContainerIPduHeaderTypeEnum set to noHeader) may be a dynamic length PDU (i.e, a PDU that at runtime may exhibit a length different from the one statically configured via Pdu.length of the respective Pdu). All other contained PDUs of a ContainerIPdu with static container layout have to be static length PDUs.

Adapt the following spec items to:

[TPS_SYST_02098] Header id and header type of a contained IPdu
A contained IPdu shall always have the same headerId per header type (long or

short header), regardless in which ContainerIPdu it is collected. If noHeader is set then the contained Ipdu does not need to have a headerId.

[TPS_SYST_02100] Relation between ContainerIPdu and contained IPdus on receiver side
On receiver side, it is not necessarily required to statically define which IPdus may be contained inside a ContainerIPdu if the header mode is used. Thus it would be possible to update the senders of ContainerIPdus and put different or additional IPdus inside.

Rework descriptions of the following classes and attributes in the model:

ContainerIPdu class description: Allows to collect several IPdus in one ContainerIPdu based on the headerType
ContainerIPdu.headerType attribute description: Defines whether and which header type is used (header id and length).

==========
Proposed solution SRS_IpduM:

- rename Chapter 6.1.4 to Dynamic I-PDU to Container Mapping
rename [SRS_IpduM_02820] to Dynamic I-PDU Mapping
add a dependency: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_0XXXX) but no mixture.
- add new Chapter Static I-PDU to Container Mapping
- add new SRS requirement: Static I-PDU Mapping
Type: Draft
Description: Multiple I-PDUs shall be mappable to a static position into a Container-PDU. If an I-PDU is mapped statically to a Container it is always transported in the same position of the Container-PDU.
Rationale: Support scenarios of Container Mapping where no header shall be used to reduce bandwidth in case always all contained PDU shall be sent.
Use Case: Efficient transmission of small PDUs on high bandwidth busses
Dependencies: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_02820) but no mixture.
–Last change on issue 76543 comment 100–

BW-C-Level:

Application	Specification	Bus
1	2	1

1.34 Specification Item SWS_IpduM_00234

Trace References:

SRS_IpduM_02825

Content:

For Container PDUs with static container layout and IpduMContainerTxTriggerMode is set to IPDUM_DIRECT, the IpduM shall trigger the Container PDU when all contained I-PDUs were updated by the upper layer.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76543: [IpduM] Container Pdu not usable for CAN-FD

  Problem description:

  With the CAN-FD extension, the Container PDU concept has been introduced. Unfortunately this it not usable in a meaningful way for CAN-FD due to the 4-byte header for each contained PDU of a Container Pdu.

  Agreed solution:

  Proposed solution for the SWS IPduM:

  Extend the enumeration of IpduMContainerHeaderSize by:
  - IPDUM_HEADERTYPE_LONG
  - IPDUM_HEADERTYPE_SHORT
  - IPDUM_HEADERTYPE_NONE

  The detailed proposed solution is contained here: https://svn.autosar.org/repos/work/09_WorkPackages/WP-A2/70_InternalDocuments/RfCs/76543-AUTOSAR_SWS_IPDUMultiplexer.doc

  ==========
  Proposed solution TPS_SystemTemplate:

  Extend the enumeration of ContainerIPduHeaderTypeEnum by:
  - noHeader: No Header is used and the the location of each containedPdu in the ContainerPdu is statically configured.

  Add the following optional attributes to ContainedIPduProps
  - offset (PositiveInteger): Byte offset that describes the location of the ContainedPdu

in the ContainerPdu if no header is used.
- updateIndicationBitPosition (PositiveInteger): The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated.

Add the following optional attributes to ContainerIPdu
- unusedBitPattern (PositiveInteger): IPduM fills not updated areas of the ContainerPdu with this byte-pattern.

Rewrite the Introduction of chapter 6.3.1 ContainerIPdu

For each IPdu which is put inside a ContainerIPdu, a header may be provided which determines which IPdu is contained (ContainedIPduProps.headerIdLongHeader or headerIdShortHeader and what the size of that IPdu is (DLC during runtime). With this header mode the receivers are able to extract the individual contained IPdus again. As an alternative option to the usage of headers a statically configured layout of IPdus in the ContainerIPdu is supported.

Introduce the following constraints:

[constr_xxxx1] Mandatory offset if noHeader is used
For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu.containedPduTriggering with ContainerIPdu.headerType = noHeader the IPdu.containedIPduProps.offset shall be defined.

[constr_xxxx2] Usage of ContainerIPdu.rxAcceptContainedIPdu if noHeader is used
If the ContainerIPdu.headerType is set to noHeader then the ContainerIPdu.rxAcceptContainedIPdu attribute value shall be set to acceptConfigured.

[constr_xxxx3] Usage of ContainedIPduProps.updateIndicationBitPosition
ContainedIPduProps.updateIndicationBitPosition is only allowed to be set to a value if the headerType of the ContainerIPdu that contains the IPdu with containedIPduProps is set to noHeader.

[constr_xxxx4] Only the last contained IPdu (according to the ContainedIPduProps.offset) of a ContainerIPdu with static container layout (i.e., a ContainerIPdu with ContainerIPduHeaderTypeEnum set to noHeader) may be a dynamic length PDU (i.e, a PDU that at runtime may exhibit a length different from the one statically configured via Pdu.length of the respective Pdu). All other contained PDUs of a ContainerIPdu with static container layout have to be static length PDUs.

Adapt the following spec items to:

[TPS_SYST_02098] Header id and header type of a contained IPdu
A contained IPdu shall always have the same headerId per header type (long or short header), regardless in which ContainerIPdu it is collected. If noHeader is set then the contained Ipdu does not need to have a headerId.

[TPS_SYST_02100] Relation between ContainerIPdu and contained IPdus on receiver side
On receiver side, it is not necessarily required to statically define which IPdus may be contained inside a ContainerIPdu if the header mode is used. Thus it would be possible to update the senders of ContainerIPdus and put different or additional IPdus inside.

Rework descriptions of the following classes and attributes in the model:

ContainerIPdu class description: Allows to collect several IPdus in one ContainerIPdu based on the headerType
ContainerIPdu.headerType attribute description: Defines whether and which header type is used (header id and length).

==========
Proposed solution SRS_IpduM:

- rename Chapter 6.1.4 to Dynamic I-PDU to Container Mapping
rename [SRS_IpduM_02820] to Dynamic I-PDU Mapping
add a dependency: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_0XXXX) but no mixture.
- add new Chapter Static I-PDU to Container Mapping
- add new SRS requirement: Static I-PDU Mapping
Type: Draft
Description: Multiple I-PDUs shall be mappable to a static position into a Container-PDU. If an I-PDU is mapped statically to a Container it is always transported in the same position of the Container-PDU.
Rationale: Support scenarios of Container Mapping where no header shall be used to reduce bandwidth in case always all contained PDU shall be sent.
Use Case: Efficient transmission of small PDUs on high bandwidth busses
Dependencies: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_02820) but no mixture.
–Last change on issue 76543 comment 100–

BW-C-Level:

Application	Specification	Bus
1	2	1

1.35 Specification Item SWS_IpduM_00235

Trace References:

SRS_IpduM_02825

Content:

In case a contained I-PDU has a configured IpduMUpdateBitPosition, the IpduM shall ensure that the update bit of this contained I-PDU is set if and only if the contained I-PDU was successfully updated.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76543: [IpduM] Container Pdu not usable for CAN-FD

  Problem description:

  With the CAN-FD extension, the Container PDU concept has been introduced. Unfortunately this it not usable in a meaningful way for CAN-FD due to the 4-byte header for each contained PDU of a Container Pdu.

  Agreed solution:

  Proposed solution for the SWS IPduM:

  Extend the enumeration of IpduMContainerHeaderSize by:
  - IPDUM_HEADERTYPE_LONG
  - IPDUM_HEADERTYPE_SHORT
  - IPDUM_HEADERTYPE_NONE

  The detailed proposed solution is contained here: https://svn.autosar.org/repos/work/09_WorkPackages/WP-A2/70_InternalDocuments/RfCs/76543-AUTOSAR_SWS_IPDUMultiplexer.doc

  ==========
  Proposed solution TPS_SystemTemplate:

Extend the enumeration of ContainerIPduHeaderTypeEnum by:
- noHeader: No Header is used and the the location of each containedPdu in the ContainerPdu is statically configured.

Add the following optional attributes to ContainedIPduProps
- offset (PositiveInteger): Byte offset that describes the location of the ContainedPdu in the ContainerPdu if no header is used.
- updateIndicationBitPosition (PositiveInteger): The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated.

Add the following optional attributes to ContainerIPdu
- unusedBitPattern (PositiveInteger): IPduM fills not updated areas of the ContainerPdu with this byte-pattern.

Rewrite the Introduction of chapter 6.3.1 ContainerIPdu

For each IPdu which is put inside a ContainerIPdu, a header may be provided which determines which IPdu is contained (ContainedIPduProps.headerIdLongHeader or headerIdShortHeader and what the size of that IPdu is (DLC during runtime). With this header mode the receivers are able to extract the individual contained IPdus again. As an alternative option to the usage of headers a statically configured layout of IPdus in the ContainerIPdu is supported.

Introduce the following constraints:

[constr_xxxx1] Mandatory offset if noHeader is used
For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu.containedPduTriggering with ContainerIPdu.headerType = noHeader the IPdu.containedIPduProps.offset shall be defined.

[constr_xxxx2] Usage of ContainerIPdu.rxAcceptContainedIPdu if noHeader is used
If the ContainerIPdu.headerType is set to noHeader then the ContainerIPdu.rxAcceptContainedIPdu attribute value shall be set to acceptConfigured.

[constr_xxxx3] Usage of ContainedIPduProps.updateIndicationBitPosition
ContainedIPduProps.updateIndicationBitPosition is only allowed to be set to a value if the headerType of the ContainerIPdu that contains the IPdu with containedIPduProps is set to noHeader.

[constr_xxxx4] Only the last contained IPdu (according to the Con-

tainedIPduProps.offset) of a ContainerIPdu with static container layout (i.e., a ContainerIPdu with ContainerIPduHeaderTypeEnum set to noHeader) may be a dynamic length PDU (i.e, a PDU that at runtime may exhibit a length different from the one statically configured via Pdu.length of the respective Pdu). All other contained PDUs of a ContainerIPdu with static container layout have to be static length PDUs.

Adapt the following spec items to:

[TPS_SYST_02098] Header id and header type of a contained IPdu
A contained IPdu shall always have the same headerId per header type (long or short header), regardless in which ContainerIPdu it is collected. If noHeader is set then the contained Ipdu does not need to have a headerId.

[TPS_SYST_02100] Relation between ContainerIPdu and contained IPdus on receiver side
On receiver side, it is not necessarily required to statically define which IPdus may be contained inside a ContainerIPdu if the header mode is used. Thus it would be possible to update the senders of ContainerIPdus and put different or additional IPdus inside.

Rework descriptions of the following classes and attributes in the model:

ContainerIPdu class description: Allows to collect several IPdus in one ContainerIPdu based on the headerType
ContainerIPdu.headerType attribute description: Defines whether and which header type is used (header id and length).

==========
Proposed solution SRS_IpduM:

- rename Chapter 6.1.4 to Dynamic I-PDU to Container Mapping
rename [SRS_IpduM_02820] to Dynamic I-PDU Mapping
add a dependency: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_0XXXX) but no mixture.
- add new Chapter Static I-PDU to Container Mapping
- add new SRS requirement: Static I-PDU Mapping
Type: Draft
Description: Multiple I-PDUs shall be mappable to a static position into a Container-PDU. If an I-PDU is mapped statically to a Container it is always transported in the same position of the Container-PDU.

Rationale: Support scenarios of Container Mapping where no header shall be used to reduce bandwidth in case always all contained PDU shall be sent.
Use Case: Efficient transmission of small PDUs on high bandwidth busses
Dependencies: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_02820) but no mixture.
–Last change on issue 76543 comment 100–

BW-C-Level:

Application	Specification	Bus
1	2	1

1.36 Specification Item SWS_IpduM_00236

Trace References:

SRS_IpduM_02825

Content:

In case a received contained I-PDU has a configured update bit, the IpduM module shall only process and indicate it to the upper layer if its received update-bit is set.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76543: [IpduM] Container Pdu not usable for CAN-FD

  Problem description:

  With the CAN-FD extension, the Container PDU concept has been introduced. Unfortunately this it not usable in a meaningful way for CAN-FD due to the 4-byte header for each contained PDU of a Container Pdu.

  Agreed solution:

  Proposed solution for the SWS IPduM:

  Extend the enumeration of IpduMContainerHeaderSize by:
  - IPDUM_HEADERTYPE_LONG
  - IPDUM_HEADERTYPE_SHORT
  - IPDUM_HEADERTYPE_NONE

  The detailed proposed solution is contained here: https://svn.autosar.org/repos/work/09_WorkPackages/WP-A2/70_InternalDocuments/RfCs/76543-AUTOSAR_SWS_IPDUMultiplexer.doc

==========

Proposed solution TPS_SystemTemplate:

Extend the enumeration of ContainerIPduHeaderTypeEnum by:
- noHeader: No Header is used and the the location of each containedPdu in the ContainerPdu is statically configured.

Add the following optional attributes to ContainedIPduProps
- offset (PositiveInteger): Byte offset that describes the location of the ContainedPdu in the ContainerPdu if no header is used.
- updateIndicationBitPosition (PositiveInteger): The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated.

Add the following optional attributes to ContainerIPdu
- unusedBitPattern (PositiveInteger): IPduM fills not updated areas of the ContainerPdu with this byte-pattern.

Rewrite the Introduction of chapter 6.3.1 ContainerIPdu

For each IPdu which is put inside a ContainerIPdu, a header may be provided which determines which IPdu is contained (ContainedIPduProps.headerIdLongHeader or headerIdShortHeader and what the size of that IPdu is (DLC during runtime). With this header mode the receivers are able to extract the individual contained IPdus again. As an alternative option to the usage of headers a statically configured layout of IPdus in the ContainerIPdu is supported.

Introduce the following constraints:

[constr_xxxx1] Mandatory offset if noHeader is used
For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu.containedPduTriggering with ContainerIPdu.headerType = noHeader the IPdu.containedIPduProps.offset shall be defined.

[constr_xxxx2] Usage of ContainerIPdu.rxAcceptContainedIPdu if noHeader is used
If the ContainerIPdu.headerType is set to noHeader then the ContainerIPdu.rxAcceptContainedIPdu attribute value shall be set to acceptConfigured.

[constr_xxxx3] Usage of ContainedIPduProps.updateIndicationBitPosition

ContainedIPduProps.updateIndicationBitPosition is only allowed to be set to a value if the headerType of the ContainerIPdu that contains the IPdu with containedIPduProps is set to noHeader.

[constr_xxxx4] Only the last contained IPdu (according to the ContainedIPduProps.offset) of a ContainerIPdu with static container layout (i.e., a ContainerIPdu with ContainerIPduHeaderTypeEnum set to noHeader) may be a dynamic length PDU (i.e, a PDU that at runtime may exhibit a length different from the one statically configured via Pdu.length of the respective Pdu). All other contained PDUs of a ContainerIPdu with static container layout have to be static length PDUs.

Adapt the following spec items to:

[TPS_SYST_02098] Header id and header type of a contained IPdu
A contained IPdu shall always have the same headerId per header type (long or short header), regardless in which ContainerIPdu it is collected. If noHeader is set then the contained Ipdu does not need to have a headerId.

[TPS_SYST_02100] Relation between ContainerIPdu and contained IPdus on receiver side
On receiver side, it is not necessarily required to statically define which IPdus may be contained inside a ContainerIPdu if the header mode is used. Thus it would be possible to update the senders of ContainerIPdus and put different or additional IPdus inside.

Rework descriptions of the following classes and attributes in the model:

ContainerIPdu class description: Allows to collect several IPdus in one ContainerIPdu based on the headerType
ContainerIPdu.headerType attribute description: Defines whether and which header type is used (header id and length).

==========
Proposed solution SRS_IpduM:

- rename Chapter 6.1.4 to Dynamic I-PDU to Container Mapping
rename [SRS_IpduM_02820] to Dynamic I-PDU Mapping
add a dependency: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_0XXXX) but no mixture.
- add new Chapter Static I-PDU to Container Mapping

- add new SRS requirement: Static I-PDU Mapping
Type: Draft
Description: Multiple I-PDUs shall be mappable to a static position into a Container-PDU. If an I-PDU is mapped statically to a Container it is always transported in the same position of the Container-PDU.
Rationale: Support scenarios of Container Mapping where no header shall be used to reduce bandwidth in case always all contained PDU shall be sent.
Use Case: Efficient transmission of small PDUs on high bandwidth busses
Dependencies: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_02820) but no mixture.
–Last change on issue 76543 comment 100–

BW-C-Level:

Application	Specification	Bus
1	2	1

1.37 Specification Item SWS_IpduM_00237

Trace References:

SRS_IpduM_02825

Content:

When processing a received Container PDU with IpduMContainerHeaderSize set to IPDUM_HEADERTYPE_NONE, the IpduM shall ignore all contained PDUs that are according to their configuration not or not completely contained in the received Container PDU. Such contained I-PDUs shall not be indicated to the upper layer. If Development Error Detection is configured (ECUC_IpduM_00132) IPDUM_E_CONTAINER shall be reported to DET via Det_ReportError.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76543: [IpduM] Container Pdu not usable for CAN-FD

  Problem description:

  With the CAN-FD extension, the Container PDU concept has been introduced. Unfortunately this it not usable in a meaningful way for CAN-FD due to the 4-byte header for each contained PDU of a Container Pdu.

  Agreed solution:

  Proposed solution for the SWS IPduM:

Extend the enumeration of IpduMContainerHeaderSize by:
- IPDUM_HEADERTYPE_LONG
- IPDUM_HEADERTYPE_SHORT
- IPDUM_HEADERTYPE_NONE

The detailed proposed solution is contained here: https://svn.autosar.org/repos/work/09_WorkPackages/WP-A2/70_InternalDocuments/RfCs/76543-AUTOSAR_SWS_IPDUMultiplexer.doc

==========
Proposed solution TPS_SystemTemplate:

Extend the enumeration of ContainerIPduHeaderTypeEnum by:
- noHeader: No Header is used and the the location of each containedPdu in the ContainerPdu is statically configured.

Add the following optional attributes to ContainedIPduProps
- offset (PositiveInteger): Byte offset that describes the location of the ContainedPdu in the ContainerPdu if no header is used.
- updateIndicationBitPosition (PositiveInteger): The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated.

Add the following optional attributes to ContainerIPdu
- unusedBitPattern (PositiveInteger): IPduM fills not updated areas of the ContainerPdu with this byte-pattern.

Rewrite the Introduction of chapter 6.3.1 ContainerIPdu

For each IPdu which is put inside a ContainerIPdu, a header may be provided which determines which IPdu is contained (ContainedIPduProps.headerIdLongHeader or headerIdShortHeader and what the size of that IPdu is (DLC during runtime). With this header mode the receivers are able to extract the individual contained IPdus again. As an alternative option to the usage of headers a statically configured layout of IPdus in the ContainerIPdu is supported.

Introduce the following constraints:

[constr_xxxx1] Mandatory offset if noHeader is used
For each IPdu which is assigned to a ContainerIPdu in the role Container-

IPdu.containedPduTriggering with ContainerIPdu.headerType = noHeader the IPdu.containedIPduProps.offset shall be defined.

[constr_xxxx2] Usage of ContainerIPdu.rxAcceptContainedIPdu if noHeader is used
If the ContainerIPdu.headerType is set to noHeader then the ContainerIPdu.rxAcceptContainedIPdu attribute value shall be set to acceptConfigured.

[constr_xxxx3] Usage of ContainedIPduProps.updateIndicationBitPosition
ContainedIPduProps.updateIndicationBitPosition is only allowed to be set to a value if the headerType of the ContainerIPdu that contains the IPdu with containedIPduProps is set to noHeader.

[constr_xxxx4] Only the last contained IPdu (according to the ContainedIPduProps.offset) of a ContainerIPdu with static container layout (i.e., a ContainerIPdu with ContainerIPduHeaderTypeEnum set to noHeader) may be a dynamic length PDU (i.e, a PDU that at runtime may exhibit a length different from the one statically configured via Pdu.length of the respective Pdu). All other contained PDUs of a ContainerIPdu with static container layout have to be static length PDUs.

Adapt the following spec items to:

[TPS_SYST_02098] Header id and header type of a contained IPdu
A contained IPdu shall always have the same headerId per header type (long or short header), regardless in which ContainerIPdu it is collected. If noHeader is set then the contained Ipdu does not need to have a headerId.

[TPS_SYST_02100] Relation between ContainerIPdu and contained IPdus on receiver side
On receiver side, it is not necessarily required to statically define which IPdus may be contained inside a ContainerIPdu if the header mode is used. Thus it would be possible to update the senders of ContainerIPdus and put different or additional IPdus inside.

Rework descriptions of the following classes and attributes in the model:

ContainerIPdu class description: Allows to collect several IPdus in one ContainerIPdu based on the headerType
ContainerIPdu.headerType attribute description: Defines whether and which header type is used (header id and length).

==========

Proposed solution SRS_IpduM:

- rename Chapter 6.1.4 to Dynamic I-PDU to Container Mapping
rename [SRS_IpduM_02820] to Dynamic I-PDU Mapping
add a dependency: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_0XXXX) but no mixture.
- add new Chapter Static I-PDU to Container Mapping
- add new SRS requirement: Static I-PDU Mapping
Type: Draft
Description: Multiple I-PDUs shall be mappable to a static position into a Container-PDU. If an I-PDU is mapped statically to a Container it is always transported in the same position of the Container-PDU.
Rationale: Support scenarios of Container Mapping where no header shall be used to reduce bandwidth in case always all contained PDU shall be sent.
Use Case: Efficient transmission of small PDUs on high bandwidth busses
Dependencies: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_02820) but no mixture.
–Last change on issue 76543 comment 100–

BW-C-Level:

Application	Specification	Bus
1	2	1

1.38 Specification Item SWS_IpduM_00238

Trace References:

SRS_IpduM_02825

Content:

For a Container PDU with IpduMContainerHeaderSize set to IPDUM_HEADERTYPE_NONE, all contained I-PDUs shall have IpduMContainedTxPduCollectionSemantics set to IPDUM_COLLECT_LAST_IS_BEST.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76543: [IpduM] Container Pdu not usable for CAN-FD

  Problem description:

With the CAN-FD extension, the Container PDU concept has been introduced. Unfortunately this it not usable in a meaningful way for CAN-FD due to the 4-byte header for each contained PDU of a Container Pdu.

Agreed solution:

Proposed solution for the SWS IPduM:

Extend the enumeration of IpduMContainerHeaderSize by:
- IPDUM_HEADERTYPE_LONG
- IPDUM_HEADERTYPE_SHORT
- IPDUM_HEADERTYPE_NONE

The detailed proposed solution is contained here: https://svn.autosar.org/repos/work/09_WorkPackages/WP-A2/70_InternalDocuments/RfCs/76543-AUTOSAR_SWS_IPDUMultiplexer.doc

==========
Proposed solution TPS_SystemTemplate:

Extend the enumeration of ContainerIPduHeaderTypeEnum by:
- noHeader: No Header is used and the the location of each containedPdu in the ContainerPdu is statically configured.

Add the following optional attributes to ContainedIPduProps
- offset (PositiveInteger): Byte offset that describes the location of the ContainedPdu in the ContainerPdu if no header is used.
- updateIndicationBitPosition (PositiveInteger): The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated.

Add the following optional attributes to ContainerIPdu
- unusedBitPattern (PositiveInteger): IPduM fills not updated areas of the ContainerPdu with this byte-pattern.

Rewrite the Introduction of chapter 6.3.1 ContainerIPdu

For each IPdu which is put inside a ContainerIPdu, a header may be provided which determines which IPdu is contained (ContainedIPduProps.headerIdLongHeader or headerIdShortHeader and what the size of that IPdu is (DLC during runtime). With this header mode the receivers are able to extract the individual contained IPdus

again. As an alternative option to the usage of headers a statically configured layout of IPdus in the ContainerIPdu is supported.

Introduce the following constraints:

[constr_xxxx1] Mandatory offset if noHeader is used
For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu.containedPduTriggering with ContainerIPdu.headerType = noHeader the IPdu.containedIPduProps.offset shall be defined.

[constr_xxxx2] Usage of ContainerIPdu.rxAcceptContainedIPdu if noHeader is used
If the ContainerIPdu.headerType is set to noHeader then the ContainerIPdu.rxAcceptContainedIPdu attribute value shall be set to acceptConfigured.

[constr_xxxx3] Usage of ContainedIPduProps.updateIndicationBitPosition
ContainedIPduProps.updateIndicationBitPosition is only allowed to be set to a value if the headerType of the ContainerIPdu that contains the IPdu with containedIPduProps is set to noHeader.

[constr_xxxx4] Only the last contained IPdu (according to the ContainedIPduProps.offset) of a ContainerIPdu with static container layout (i.e., a ContainerIPdu with ContainerIPduHeaderTypeEnum set to noHeader) may be a dynamic length PDU (i.e, a PDU that at runtime may exhibit a length different from the one statically configured via Pdu.length of the respective Pdu). All other contained PDUs of a ContainerIPdu with static container layout have to be static length PDUs.

Adapt the following spec items to:

[TPS_SYST_02098] Header id and header type of a contained IPdu
A contained IPdu shall always have the same headerId per header type (long or short header), regardless in which ContainerIPdu it is collected. If noHeader is set then the contained Ipdu does not need to have a headerId.

[TPS_SYST_02100] Relation between ContainerIPdu and contained IPdus on receiver side
On receiver side, it is not necessarily required to statically define which IPdus may be contained inside a ContainerIPdu if the header mode is used. Thus it would be possible to update the senders of ContainerIPdus and put different or additional IPdus inside.

Rework descriptions of the following classes and attributes in the model:

ContainerIPdu class description: Allows to collect several IPdus in one ContainerIPdu based on the headerType
ContainerIPdu.headerType attribute description: Defines whether and which header type is used (header id and length).

==========
Proposed solution SRS_IpduM:

- rename Chapter 6.1.4 to Dynamic I-PDU to Container Mapping
rename [SRS_IpduM_02820] to Dynamic I-PDU Mapping
add a dependency: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_0XXXX) but no mixture.
- add new Chapter Static I-PDU to Container Mapping
- add new SRS requirement: Static I-PDU Mapping
Type: Draft
Description: Multiple I-PDUs shall be mappable to a static position into a Container-PDU. If an I-PDU is mapped statically to a Container it is always transported in the same position of the Container-PDU.
Rationale: Support scenarios of Container Mapping where no header shall be used to reduce bandwidth in case always all contained PDU shall be sent.
Use Case: Efficient transmission of small PDUs on high bandwidth busses
Dependencies: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_02820) but no mixture.
–Last change on issue 76543 comment 100–

BW-C-Level:

Application	Specification	Bus
1	2	1

1.39 Specification Item SWS_IpduM_00240

Trace References:

SRS_IpduM_02825

Content:

Contained I-PDUs with a configured IpduMPduUpdateBitPosition shall only be assigned to Container PDUs with IpduMContainerHeaderSize set to IPDUM_HEADERTYPE_NONE.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76543: [IpduM] Container Pdu not usable for CAN-FD

  Problem description:

  With the CAN-FD extension, the Container PDU concept has been introduced. Unfortunately this it not usable in a meaningful way for CAN-FD due to the 4-byte header for each contained PDU of a Container Pdu.

  Agreed solution:

  Proposed solution for the SWS IPduM:

  Extend the enumeration of IpduMContainerHeaderSize by:
  - IPDUM_HEADERTYPE_LONG
  - IPDUM_HEADERTYPE_SHORT
  - IPDUM_HEADERTYPE_NONE

  The detailed proposed solution is contained here: https://svn.autosar.org/repos/work/09_WorkPackages/WP-A2/70_InternalDocuments/RfCs/76543-AUTOSAR_SWS_IPDUMultiplexer.doc

  ==========
  Proposed solution TPS_SystemTemplate:

  Extend the enumeration of ContainerIPduHeaderTypeEnum by:
  - noHeader: No Header is used and the the location of each containedPdu in the ContainerPdu is statically configured.

  Add the following optional attributes to ContainedIPduProps
  - offset (PositiveInteger): Byte offset that describes the location of the ContainedPdu in the ContainerPdu if no header is used.
  - updateIndicationBitPosition (PositiveInteger): The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated.

  Add the following optional attributes to ContainerIPdu
  - unusedBitPattern (PositiveInteger): IPduM fills not updated areas of the ContainerPdu with this byte-pattern.

  Rewrite the Introduction of chapter 6.3.1 ContainerIPdu

For each IPdu which is put inside a ContainerIPdu, a header may be provided which determines which IPdu is contained (ContainedIPduProps.headerIdLongHeader or headerIdShortHeader and what the size of that IPdu is (DLC during runtime). With this header mode the receivers are able to extract the individual contained IPdus again. As an alternative option to the usage of headers a statically configured layout of IPdus in the ContainerIPdu is supported.

Introduce the following constraints:

[constr_xxxx1] Mandatory offset if noHeader is used
For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu.containedPduTriggering with ContainerIPdu.headerType = noHeader the IPdu.containedIPduProps.offset shall be defined.

[constr_xxxx2] Usage of ContainerIPdu.rxAcceptContainedIPdu if noHeader is used
If the ContainerIPdu.headerType is set to noHeader then the ContainerIPdu.rxAcceptContainedIPdu attribute value shall be set to acceptConfigured.

[constr_xxxx3] Usage of ContainedIPduProps.updateIndicationBitPosition
ContainedIPduProps.updateIndicationBitPosition is only allowed to be set to a value if the headerType of the ContainerIPdu that contains the IPdu with containedIPduProps is set to noHeader.

[constr_xxxx4] Only the last contained IPdu (according to the ContainedIPduProps.offset) of a ContainerIPdu with static container layout (i.e., a ContainerIPdu with ContainerIPduHeaderTypeEnum set to noHeader) may be a dynamic length PDU (i.e, a PDU that at runtime may exhibit a length different from the one statically configured via Pdu.length of the respective Pdu). All other contained PDUs of a ContainerIPdu with static container layout have to be static length PDUs.

Adapt the following spec items to:

[TPS_SYST_02098] Header id and header type of a contained IPdu
A contained IPdu shall always have the same headerId per header type (long or short header), regardless in which ContainerIPdu it is collected. If noHeader is set then the contained Ipdu does not need to have a headerId.

[TPS_SYST_02100] Relation between ContainerIPdu and contained IPdus on receiver side
On receiver side, it is not necessarily required to statically define which IPdus may

be contained inside a ContainerIPdu if the header mode is used. Thus it would be possible to update the senders of ContainerIPdus and put different or additional IPdus inside.

Rework descriptions of the following classes and attributes in the model:

ContainerIPdu class description: Allows to collect several IPdus in one ContainerIPdu based on the headerType
ContainerIPdu.headerType attribute description: Defines whether and which header type is used (header id and length).

==========
Proposed solution SRS_IpduM:

- rename Chapter 6.1.4 to Dynamic I-PDU to Container Mapping
rename [SRS_IpduM_02820] to Dynamic I-PDU Mapping
add a dependency: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_0XXXX) but no mixture.
- add new Chapter Static I-PDU to Container Mapping
- add new SRS requirement: Static I-PDU Mapping
Type: Draft
Description: Multiple I-PDUs shall be mappable to a static position into a Container-PDU. If an I-PDU is mapped statically to a Container it is always transported in the same position of the Container-PDU.
Rationale: Support scenarios of Container Mapping where no header shall be used to reduce bandwidth in case always all contained PDU shall be sent.
Use Case: Efficient transmission of small PDUs on high bandwidth busses
Dependencies: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_02820) but no mixture.
–Last change on issue 76543 comment 100–

BW-C-Level:

Application	Specification	Bus
1	2	1

1.40 Specification Item SWS_IpduM_00241

Trace References:

SRS_IpduM_02825

Content:

For a Container PDU with IpduMContainerHeaderSize set to IPDUM_HEADERTYPE_NONE, all contained I-PDUs shall have a configured IpduMContainedTxPduOffset.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76543: [IpduM] Container Pdu not usable for CAN-FD

  Problem description:

  With the CAN-FD extension, the Container PDU concept has been introduced. Unfortunately this it not usable in a meaningful way for CAN-FD due to the 4-byte header for each contained PDU of a Container Pdu.

  Agreed solution:

  Proposed solution for the SWS IPduM:

  Extend the enumeration of IpduMContainerHeaderSize by:
  - IPDUM_HEADERTYPE_LONG
  - IPDUM_HEADERTYPE_SHORT
  - IPDUM_HEADERTYPE_NONE

  The detailed proposed solution is contained here: https://svn.autosar.org/repos/work/09_WorkPackages/WP-A2/70_InternalDocuments/RfCs/76543-AUTOSAR_SWS_IPDUMultiplexer.doc

  ==========
  Proposed solution TPS_SystemTemplate:

  Extend the enumeration of ContainerIPduHeaderTypeEnum by:
  - noHeader: No Header is used and the the location of each containedPdu in the ContainerPdu is statically configured.

  Add the following optional attributes to ContainedIPduProps
  - offset (PositiveInteger): Byte offset that describes the location of the ContainedPdu in the ContainerPdu if no header is used.
  - updateIndicationBitPosition (PositiveInteger): The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated.

Add the following optional attributes to ContainerIPdu
- unusedBitPattern (PositiveInteger): IPduM fills not updated areas of the ContainerPdu with this byte-pattern.

Rewrite the Introduction of chapter 6.3.1 ContainerIPdu

For each IPdu which is put inside a ContainerIPdu, a header may be provided which determines which IPdu is contained (ContainedIPduProps.headerIdLongHeader or headerIdShortHeader and what the size of that IPdu is (DLC during runtime). With this header mode the receivers are able to extract the individual contained IPdus again. As an alternative option to the usage of headers a statically configured layout of IPdus in the ContainerIPdu is supported.

Introduce the following constraints:

[constr_xxxx1] Mandatory offset if noHeader is used
For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu.containedPduTriggering with ContainerIPdu.headerType = noHeader the IPdu.containedIPduProps.offset shall be defined.

[constr_xxxx2] Usage of ContainerIPdu.rxAcceptContainedIPdu if noHeader is used
If the ContainerIPdu.headerType is set to noHeader then the ContainerIPdu.rxAcceptContainedIPdu attribute value shall be set to acceptConfigured.

[constr_xxxx3] Usage of ContainedIPduProps.updateIndicationBitPosition
ContainedIPduProps.updateIndicationBitPosition is only allowed to be set to a value if the headerType of the ContainerIPdu that contains the IPdu with containedIPduProps is set to noHeader.

[constr_xxxx4] Only the last contained IPdu (according to the ContainedIPduProps.offset) of a ContainerIPdu with static container layout (i.e., a ContainerIPdu with ContainerIPduHeaderTypeEnum set to noHeader) may be a dynamic length PDU (i.e, a PDU that at runtime may exhibit a length different from the one statically configured via Pdu.length of the respective Pdu). All other contained PDUs of a ContainerIPdu with static container layout have to be static length PDUs.

Adapt the following spec items to:

[TPS_SYST_02098] Header id and header type of a contained IPdu
A contained IPdu shall always have the same headerId per header type (long or

short header), regardless in which ContainerIPdu it is collected. If noHeader is set then the contained Ipdu does not need to have a headerId.

[TPS_SYST_02100] Relation between ContainerIPdu and contained IPdus on receiver side
On receiver side, it is not necessarily required to statically define which IPdus may be contained inside a ContainerIPdu if the header mode is used. Thus it would be possible to update the senders of ContainerIPdus and put different or additional IPdus inside.

Rework descriptions of the following classes and attributes in the model:

ContainerIPdu class description: Allows to collect several IPdus in one ContainerIPdu based on the headerType
ContainerIPdu.headerType attribute description: Defines whether and which header type is used (header id and length).

==========
Proposed solution SRS_IpduM:

- rename Chapter 6.1.4 to Dynamic I-PDU to Container Mapping
rename [SRS_IpduM_02820] to Dynamic I-PDU Mapping
add a dependency: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_0XXXX) but no mixture.
- add new Chapter Static I-PDU to Container Mapping
- add new SRS requirement: Static I-PDU Mapping
Type: Draft
Description: Multiple I-PDUs shall be mappable to a static position into a Container-PDU. If an I-PDU is mapped statically to a Container it is always transported in the same position of the Container-PDU.
Rationale: Support scenarios of Container Mapping where no header shall be used to reduce bandwidth in case always all contained PDU shall be sent.
Use Case: Efficient transmission of small PDUs on high bandwidth busses
Dependencies: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_02820) but no mixture.
–Last change on issue 76543 comment 100–

BW-C-Level:

Application	Specification	Bus
1	2	1

1.41 Specification Item SWS_IpduM_00242

Trace References:

SRS_IpduM_02825

Content:

For a Container PDU with IpduMContainerHeaderSize set to IPDUM_HEADERTYPE_NONE and IpduMUnusedAreasDefault not set, all contained I-PDUs shall have a configured IpduMPduUpdateBitPosition.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76543: [IpduM] Container Pdu not usable for CAN-FD

  Problem description:

  With the CAN-FD extension, the Container PDU concept has been introduced. Unfortunately this it not usable in a meaningful way for CAN-FD due to the 4-byte header for each contained PDU of a Container Pdu.

  Agreed solution:

  Proposed solution for the SWS IPduM:

  Extend the enumeration of IpduMContainerHeaderSize by:
  - IPDUM_HEADERTYPE_LONG
  - IPDUM_HEADERTYPE_SHORT
  - IPDUM_HEADERTYPE_NONE

  The detailed proposed solution is contained here: https://svn.autosar.org/repos/work/09_WorkPackages/WP-A2/70_InternalDocuments/RfCs/76543-AUTOSAR_SWS_IPDUMultiplexer.doc

  ==========
  Proposed solution TPS_SystemTemplate:

  Extend the enumeration of ContainerIPduHeaderTypeEnum by:
  - noHeader: No Header is used and the the location of each containedPdu in the ContainerPdu is statically configured.

  Add the following optional attributes to ContainedIPduProps
  - offset (PositiveInteger): Byte offset that describes the location of the ContainedPdu

in the ContainerPdu if no header is used.
- updateIndicationBitPosition (PositiveInteger): The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated.

Add the following optional attributes to ContainerIPdu
- unusedBitPattern (PositiveInteger): IPduM fills not updated areas of the ContainerPdu with this byte-pattern.

Rewrite the Introduction of chapter 6.3.1 ContainerIPdu

For each IPdu which is put inside a ContainerIPdu, a header may be provided which determines which IPdu is contained (ContainedIPduProps.headerIdLongHeader or headerIdShortHeader and what the size of that IPdu is (DLC during runtime). With this header mode the receivers are able to extract the individual contained IPdus again. As an alternative option to the usage of headers a statically configured layout of IPdus in the ContainerIPdu is supported.

Introduce the following constraints:

[constr_xxxx1] Mandatory offset if noHeader is used
For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu.containedPduTriggering with ContainerIPdu.headerType = noHeader the IPdu.containedIPduProps.offset shall be defined.

[constr_xxxx2] Usage of ContainerIPdu.rxAcceptContainedIPdu if noHeader is used
If the ContainerIPdu.headerType is set to noHeader then the ContainerIPdu.rxAcceptContainedIPdu attribute value shall be set to acceptConfigured.

[constr_xxxx3] Usage of ContainedIPduProps.updateIndicationBitPosition
ContainedIPduProps.updateIndicationBitPosition is only allowed to be set to a value if the headerType of the ContainerIPdu that contains the IPdu with containedIPduProps is set to noHeader.

[constr_xxxx4] Only the last contained IPdu (according to the ContainedIPduProps.offset) of a ContainerIPdu with static container layout (i.e., a ContainerIPdu with ContainerIPduHeaderTypeEnum set to noHeader) may be a dynamic length PDU (i.e, a PDU that at runtime may exhibit a length different from the one statically configured via Pdu.length of the respective Pdu). All other contained PDUs of a ContainerIPdu with static container layout have to be static length PDUs.

Adapt the following spec items to:

[TPS_SYST_02098] Header id and header type of a contained IPdu
A contained IPdu shall always have the same headerId per header type (long or short header), regardless in which ContainerIPdu it is collected. If noHeader is set then the contained Ipdu does not need to have a headerId.

[TPS_SYST_02100] Relation between ContainerIPdu and contained IPdus on receiver side
On receiver side, it is not necessarily required to statically define which IPdus may be contained inside a ContainerIPdu if the header mode is used. Thus it would be possible to update the senders of ContainerIPdus and put different or additional IPdus inside.

Rework descriptions of the following classes and attributes in the model:

ContainerIPdu class description: Allows to collect several IPdus in one ContainerIPdu based on the headerType
ContainerIPdu.headerType attribute description: Defines whether and which header type is used (header id and length).

==========
Proposed solution SRS_IpduM:

- rename Chapter 6.1.4 to Dynamic I-PDU to Container Mapping
rename [SRS_IpduM_02820] to Dynamic I-PDU Mapping
add a dependency: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_0XXXX) but no mixture.
- add new Chapter Static I-PDU to Container Mapping
- add new SRS requirement: Static I-PDU Mapping
Type: Draft
Description: Multiple I-PDUs shall be mappable to a static position into a Container-PDU. If an I-PDU is mapped statically to a Container it is always transported in the same position of the Container-PDU.
Rationale: Support scenarios of Container Mapping where no header shall be used to reduce bandwidth in case always all contained PDU shall be sent.
Use Case: Efficient transmission of small PDUs on high bandwidth busses
Dependencies: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_02820) but no mixture.
–Last change on issue 76543 comment 100–

BW-C-Level:

Application	Specification	Bus
1	2	1

1.42 Specification Item SWS_IpduM_00243

Trace References:

SRS_IpduM_02825

Content:

When sending a Static Container PDU, the IpduM shall trim the size of the Container PDU if possible to avoid transmission of not updated data. For this purpose, not updated contained I-PDUs and not updated areas of dynamic length PDUs shall be cut off.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76543: [IpduM] Container Pdu not usable for CAN-FD

  Problem description:

  With the CAN-FD extension, the Container PDU concept has been introduced. Unfortunately this it not usable in a meaningful way for CAN-FD due to the 4-byte header for each contained PDU of a Container Pdu.

  Agreed solution:

  Proposed solution for the SWS IPduM:

  Extend the enumeration of IpduMContainerHeaderSize by:
  - IPDUM_HEADERTYPE_LONG
  - IPDUM_HEADERTYPE_SHORT
  - IPDUM_HEADERTYPE_NONE

  The detailed proposed solution is contained here: https://svn.autosar.org/repos/work/09_WorkPackages/WP-A2/70_InternalDocuments/RfCs/76543-AUTOSAR_SWS_IPDUMultiplexer.doc

  ==========
  Proposed solution TPS_SystemTemplate:

Extend the enumeration of ContainerIPduHeaderTypeEnum by:
- noHeader: No Header is used and the the location of each containedPdu in the ContainerPdu is statically configured.

Add the following optional attributes to ContainedIPduProps
- offset (PositiveInteger): Byte offset that describes the location of the ContainedPdu in the ContainerPdu if no header is used.
- updateIndicationBitPosition (PositiveInteger): The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated.

Add the following optional attributes to ContainerIPdu
- unusedBitPattern (PositiveInteger): IPduM fills not updated areas of the ContainerPdu with this byte-pattern.

Rewrite the Introduction of chapter 6.3.1 ContainerIPdu

For each IPdu which is put inside a ContainerIPdu, a header may be provided which determines which IPdu is contained (ContainedIPduProps.headerIdLongHeader or headerIdShortHeader and what the size of that IPdu is (DLC during runtime). With this header mode the receivers are able to extract the individual contained IPdus again. As an alternative option to the usage of headers a statically configured layout of IPdus in the ContainerIPdu is supported.

Introduce the following constraints:

[constr_xxxx1] Mandatory offset if noHeader is used
For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu.containedPduTriggering with ContainerIPdu.headerType = noHeader the IPdu.containedIPduProps.offset shall be defined.

[constr_xxxx2] Usage of ContainerIPdu.rxAcceptContainedIPdu if noHeader is used
If the ContainerIPdu.headerType is set to noHeader then the ContainerIPdu.rxAcceptContainedIPdu attribute value shall be set to acceptConfigured.

[constr_xxxx3] Usage of ContainedIPduProps.updateIndicationBitPosition
ContainedIPduProps.updateIndicationBitPosition is only allowed to be set to a value if the headerType of the ContainerIPdu that contains the IPdu with containedIPduProps is set to noHeader.

[constr_xxxx4] Only the last contained IPdu (according to the Con-

tainedIPduProps.offset) of a ContainerIPdu with static container layout (i.e., a ContainerIPdu with ContainerIPduHeaderTypeEnum set to noHeader) may be a dynamic length PDU (i.e, a PDU that at runtime may exhibit a length different from the one statically configured via Pdu.length of the respective Pdu). All other contained PDUs of a ContainerIPdu with static container layout have to be static length PDUs.

Adapt the following spec items to:

[TPS_SYST_02098] Header id and header type of a contained IPdu
A contained IPdu shall always have the same headerId per header type (long or short header), regardless in which ContainerIPdu it is collected. If noHeader is set then the contained Ipdu does not need to have a headerId.

[TPS_SYST_02100] Relation between ContainerIPdu and contained IPdus on receiver side
On receiver side, it is not necessarily required to statically define which IPdus may be contained inside a ContainerIPdu if the header mode is used. Thus it would be possible to update the senders of ContainerIPdus and put different or additional IPdus inside.

Rework descriptions of the following classes and attributes in the model:

ContainerIPdu class description: Allows to collect several IPdus in one ContainerIPdu based on the headerType
ContainerIPdu.headerType attribute description: Defines whether and which header type is used (header id and length).

==========
Proposed solution SRS_IpduM:

- rename Chapter 6.1.4 to Dynamic I-PDU to Container Mapping
rename [SRS_IpduM_02820] to Dynamic I-PDU Mapping
add a dependency: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_0XXXX) but no mixture.
- add new Chapter Static I-PDU to Container Mapping
- add new SRS requirement: Static I-PDU Mapping
Type: Draft
Description: Multiple I-PDUs shall be mappable to a static position into a Container-PDU. If an I-PDU is mapped statically to a Container it is always transported in the same position of the Container-PDU.

Rationale: Support scenarios of Container Mapping where no header shall be used to reduce bandwidth in case always all contained PDU shall be sent.
Use Case: Efficient transmission of small PDUs on high bandwidth busses
Dependencies: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_02820) but no mixture.
–Last change on issue 76543 comment 100–

BW-C-Level:

Application	Specification	Bus
1	2	1

1.43 Specification Item SWS_IpduM_00245

Trace References:

SRS_IpduM_02825

Content:

All IpduMPduUpdateBitPositions shall be configured to their own not otherwise occupied bit position.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76543: [IpduM] Container Pdu not usable for CAN-FD

  Problem description:

  With the CAN-FD extension, the Container PDU concept has been introduced. Unfortunately this it not usable in a meaningful way for CAN-FD due to the 4-byte header for each contained PDU of a Container Pdu.

  Agreed solution:

  Proposed solution for the SWS IPduM:

  Extend the enumeration of IpduMContainerHeaderSize by:
  - IPDUM_HEADERTYPE_LONG
  - IPDUM_HEADERTYPE_SHORT
  - IPDUM_HEADERTYPE_NONE

  The detailed proposed solution is contained here: https://svn.autosar.org/repos/work/09_WorkPackages/WP-A2/70_InternalDocuments/RfCs/76543-AUTOSAR_SWS_IPDUMultiplexer.doc

==========

Proposed solution TPS_SystemTemplate:

Extend the enumeration of ContainerIPduHeaderTypeEnum by:
- noHeader: No Header is used and the the location of each containedPdu in the ContainerPdu is statically configured.

Add the following optional attributes to ContainedIPduProps
- offset (PositiveInteger): Byte offset that describes the location of the ContainedPdu in the ContainerPdu if no header is used.
- updateIndicationBitPosition (PositiveInteger): The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated.

Add the following optional attributes to ContainerIPdu
- unusedBitPattern (PositiveInteger): IPduM fills not updated areas of the ContainerPdu with this byte-pattern.

Rewrite the Introduction of chapter 6.3.1 ContainerIPdu

For each IPdu which is put inside a ContainerIPdu, a header may be provided which determines which IPdu is contained (ContainedIPduProps.headerIdLongHeader or headerIdShortHeader and what the size of that IPdu is (DLC during runtime). With this header mode the receivers are able to extract the individual contained IPdus again. As an alternative option to the usage of headers a statically configured layout of IPdus in the ContainerIPdu is supported.

Introduce the following constraints:

[constr_xxxx1] Mandatory offset if noHeader is used
For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu.containedPduTriggering with ContainerIPdu.headerType = noHeader the IPdu.containedIPduProps.offset shall be defined.

[constr_xxxx2] Usage of ContainerIPdu.rxAcceptContainedIPdu if noHeader is used
If the ContainerIPdu.headerType is set to noHeader then the ContainerIPdu.rxAcceptContainedIPdu attribute value shall be set to acceptConfigured.

[constr_xxxx3] Usage of ContainedIPduProps.updateIndicationBitPosition

ContainedIPduProps.updateIndicationBitPosition is only allowed to be set to a value if the headerType of the ContainerIPdu that contains the IPdu with containedIPduProps is set to noHeader.

[constr_xxxx4] Only the last contained IPdu (according to the ContainedIPduProps.offset) of a ContainerIPdu with static container layout (i.e., a ContainerIPdu with ContainerIPduHeaderTypeEnum set to noHeader) may be a dynamic length PDU (i.e, a PDU that at runtime may exhibit a length different from the one statically configured via Pdu.length of the respective Pdu). All other contained PDUs of a ContainerIPdu with static container layout have to be static length PDUs.

Adapt the following spec items to:

[TPS_SYST_02098] Header id and header type of a contained IPdu
A contained IPdu shall always have the same headerId per header type (long or short header), regardless in which ContainerIPdu it is collected. If noHeader is set then the contained Ipdu does not need to have a headerId.

[TPS_SYST_02100] Relation between ContainerIPdu and contained IPdus on receiver side
On receiver side, it is not necessarily required to statically define which IPdus may be contained inside a ContainerIPdu if the header mode is used. Thus it would be possible to update the senders of ContainerIPdus and put different or additional IPdus inside.

Rework descriptions of the following classes and attributes in the model:

ContainerIPdu class description: Allows to collect several IPdus in one ContainerIPdu based on the headerType
ContainerIPdu.headerType attribute description: Defines whether and which header type is used (header id and length).

==========
Proposed solution SRS_IpduM:

- rename Chapter 6.1.4 to Dynamic I-PDU to Container Mapping
rename [SRS_IpduM_02820] to Dynamic I-PDU Mapping
add a dependency: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_0XXXX) but no mixture.
- add new Chapter Static I-PDU to Container Mapping

- add new SRS requirement: Static I-PDU Mapping
Type: Draft
Description: Multiple I-PDUs shall be mappable to a static position into a Container-PDU. If an I-PDU is mapped statically to a Container it is always transported in the same position of the Container-PDU.
Rationale: Support scenarios of Container Mapping where no header shall be used to reduce bandwidth in case always all contained PDU shall be sent.
Use Case: Efficient transmission of small PDUs on high bandwidth busses
Dependencies: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_02820) but no mixture.
–Last change on issue 76543 comment 100–

BW-C-Level:

Application	Specification	Bus
1	2	1

1.44 Specification Item SWS_IpduM_00246

Trace References:

SRS_IpduM_02825

Content:

Only the last contained IPdu (according to IpduMContainedPduOffset) of a ContainerIPdu with static container layout (i.e. IpduMContainerHeaderSize set to IPDUM_HEADERTYPE_NONE) may be a dynamic length PDU (i.e, a PDU that at runtime may exhibit a length different from the one statically configured via Pdu.length of the respective Pdu). All other contained PDUs of a ContainerIPdu with static container layout have to be static length PDUs.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76543: [IpduM] Container Pdu not usable for CAN-FD

  Problem description:

  With the CAN-FD extension, the Container PDU concept has been introduced. Unfortunately this it not usable in a meaningful way for CAN-FD due to the 4-byte header for each contained PDU of a Container Pdu.

  Agreed solution:

  Proposed solution for the SWS IPduM:

Extend the enumeration of IpduMContainerHeaderSize by:
- IPDUM_HEADERTYPE_LONG
- IPDUM_HEADERTYPE_SHORT
- IPDUM_HEADERTYPE_NONE

The detailed proposed solution is contained here: https://svn.autosar.org/repos/work/09_WorkPackages/WP-A2/70_InternalDocuments/RfCs/76543-AUTOSAR_SWS_IPDUMultiplexer.doc

==========
Proposed solution TPS_SystemTemplate:

Extend the enumeration of ContainerIPduHeaderTypeEnum by:
- noHeader: No Header is used and the the location of each containedPdu in the ContainerPdu is statically configured.

Add the following optional attributes to ContainedIPduProps
- offset (PositiveInteger): Byte offset that describes the location of the ContainedPdu in the ContainerPdu if no header is used.
- updateIndicationBitPosition (PositiveInteger): The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated.

Add the following optional attributes to ContainerIPdu
- unusedBitPattern (PositiveInteger): IPduM fills not updated areas of the ContainerPdu with this byte-pattern.

Rewrite the Introduction of chapter 6.3.1 ContainerIPdu

For each IPdu which is put inside a ContainerIPdu, a header may be provided which determines which IPdu is contained (ContainedIPduProps.headerIdLongHeader or headerIdShortHeader and what the size of that IPdu is (DLC during runtime). With this header mode the receivers are able to extract the individual contained IPdus again. As an alternative option to the usage of headers a statically configured layout of IPdus in the ContainerIPdu is supported.

Introduce the following constraints:

[constr_xxxx1] Mandatory offset if noHeader is used
For each IPdu which is assigned to a ContainerIPdu in the role Container-

IPdu.containedPduTriggering with ContainerIPdu.headerType = noHeader the IPdu.containedIPduProps.offset shall be defined.

[constr_xxxx2] Usage of ContainerIPdu.rxAcceptContainedIPdu if noHeader is used
If the ContainerIPdu.headerType is set to noHeader then the ContainerIPdu.rxAcceptContainedIPdu attribute value shall be set to acceptConfigured.

[constr_xxxx3] Usage of ContainedIPduProps.updateIndicationBitPosition
ContainedIPduProps.updateIndicationBitPosition is only allowed to be set to a value if the headerType of the ContainerIPdu that contains the IPdu with containedIPduProps is set to noHeader.

[constr_xxxx4] Only the last contained IPdu (according to the ContainedIPduProps.offset) of a ContainerIPdu with static container layout (i.e., a ContainerIPdu with ContainerIPduHeaderTypeEnum set to noHeader) may be a dynamic length PDU (i.e, a PDU that at runtime may exhibit a length different from the one statically configured via Pdu.length of the respective Pdu). All other contained PDUs of a ContainerIPdu with static container layout have to be static length PDUs.

Adapt the following spec items to:

[TPS_SYST_02098] Header id and header type of a contained IPdu
A contained IPdu shall always have the same headerId per header type (long or short header), regardless in which ContainerIPdu it is collected. If noHeader is set then the contained Ipdu does not need to have a headerId.

[TPS_SYST_02100] Relation between ContainerIPdu and contained IPdus on receiver side
On receiver side, it is not necessarily required to statically define which IPdus may be contained inside a ContainerIPdu if the header mode is used. Thus it would be possible to update the senders of ContainerIPdus and put different or additional IPdus inside.

Rework descriptions of the following classes and attributes in the model:

ContainerIPdu class description: Allows to collect several IPdus in one ContainerIPdu based on the headerType
ContainerIPdu.headerType attribute description: Defines whether and which header type is used (header id and length).

==========

Proposed solution SRS_IpduM:

- rename Chapter 6.1.4 to Dynamic I-PDU to Container Mapping
rename [SRS_IpduM_02820] to Dynamic I-PDU Mapping
add a dependency: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_0XXXX) but no mixture.
- add new Chapter Static I-PDU to Container Mapping
- add new SRS requirement: Static I-PDU Mapping
Type: Draft
Description: Multiple I-PDUs shall be mappable to a static position into a Container-PDU. If an I-PDU is mapped statically to a Container it is always transported in the same position of the Container-PDU.
Rationale: Support scenarios of Container Mapping where no header shall be used to reduce bandwidth in case always all contained PDU shall be sent.
Use Case: Efficient transmission of small PDUs on high bandwidth busses
Dependencies: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_02820) but no mixture.
–Last change on issue 76543 comment 100–

BW-C-Level:

Application	Specification	Bus
1	2	1

1.45 Specification Item SWS_IpduM_00247

Trace References:

SRS_IpduM_02825

Content:

Partly or erroneous container received

- error code: IPDUM_E_CONTAINER
- value [hex]: 0x32

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76543: [IpduM] Container Pdu not usable for CAN-FD

  Problem description:

With the CAN-FD extension, the Container PDU concept has been introduced. Unfortunately this it not usable in a meaningful way for CAN-FD due to the 4-byte header for each contained PDU of a Container Pdu.

Agreed solution:

Proposed solution for the SWS IPduM:

Extend the enumeration of IpduMContainerHeaderSize by:
- IPDUM_HEADERTYPE_LONG
- IPDUM_HEADERTYPE_SHORT
- IPDUM_HEADERTYPE_NONE

The detailed proposed solution is contained here: https://svn.autosar.org/repos/work/09_WorkPackages/WP-A2/70_InternalDocuments/RfCs/76543-AUTOSAR_SWS_IPDUMultiplexer.doc

==========
Proposed solution TPS_SystemTemplate:

Extend the enumeration of ContainerIPduHeaderTypeEnum by:
- noHeader: No Header is used and the the location of each containedPdu in the ContainerPdu is statically configured.

Add the following optional attributes to ContainedIPduProps
- offset (PositiveInteger): Byte offset that describes the location of the ContainedPdu in the ContainerPdu if no header is used.
- updateIndicationBitPosition (PositiveInteger): The updateIndicationBit specifies the bit location of ContainedIPdu Update-Bit in the Container PDU. It indicates to the receivers that the ContainedIPdu in the ContainerIPdu was updated.

Add the following optional attributes to ContainerIPdu
- unusedBitPattern (PositiveInteger): IPduM fills not updated areas of the ContainerPdu with this byte-pattern.

Rewrite the Introduction of chapter 6.3.1 ContainerIPdu

For each IPdu which is put inside a ContainerIPdu, a header may be provided which determines which IPdu is contained (ContainedIPduProps.headerIdLongHeader or headerIdShortHeader and what the size of that IPdu is (DLC during runtime). With this header mode the receivers are able to extract the individual contained IPdus

again. As an alternative option to the usage of headers a statically configured layout of IPdus in the ContainerIPdu is supported.

Introduce the following constraints:

[constr_xxxx1] Mandatory offset if noHeader is used
For each IPdu which is assigned to a ContainerIPdu in the role ContainerIPdu.containedPduTriggering with ContainerIPdu.headerType = noHeader the IPdu.containedIPduProps.offset shall be defined.

[constr_xxxx2] Usage of ContainerIPdu.rxAcceptContainedIPdu if noHeader is used
If the ContainerIPdu.headerType is set to noHeader then the ContainerIPdu.rxAcceptContainedIPdu attribute value shall be set to acceptConfigured.

[constr_xxxx3] Usage of ContainedIPduProps.updateIndicationBitPosition
ContainedIPduProps.updateIndicationBitPosition is only allowed to be set to a value if the headerType of the ContainerIPdu that contains the IPdu with containedIPduProps is set to noHeader.

[constr_xxxx4] Only the last contained IPdu (according to the ContainedIPduProps.offset) of a ContainerIPdu with static container layout (i.e., a ContainerIPdu with ContainerIPduHeaderTypeEnum set to noHeader) may be a dynamic length PDU (i.e, a PDU that at runtime may exhibit a length different from the one statically configured via Pdu.length of the respective Pdu). All other contained PDUs of a ContainerIPdu with static container layout have to be static length PDUs.

Adapt the following spec items to:

[TPS_SYST_02098] Header id and header type of a contained IPdu
A contained IPdu shall always have the same headerId per header type (long or short header), regardless in which ContainerIPdu it is collected. If noHeader is set then the contained Ipdu does not need to have a headerId.

[TPS_SYST_02100] Relation between ContainerIPdu and contained IPdus on receiver side
On receiver side, it is not necessarily required to statically define which IPdus may be contained inside a ContainerIPdu if the header mode is used. Thus it would be possible to update the senders of ContainerIPdus and put different or additional IPdus inside.

Rework descriptions of the following classes and attributes in the model:

ContainerIPdu class description: Allows to collect several IPdus in one ContainerIPdu based on the headerType
ContainerIPdu.headerType attribute description: Defines whether and which header type is used (header id and length).

==========
Proposed solution SRS_IpduM:

- rename Chapter 6.1.4 to Dynamic I-PDU to Container Mapping
rename [SRS_IpduM_02820] to Dynamic I-PDU Mapping
add a dependency: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_0XXXX) but no mixture.
- add new Chapter Static I-PDU to Container Mapping
- add new SRS requirement: Static I-PDU Mapping
Type: Draft
Description: Multiple I-PDUs shall be mappable to a static position into a Container-PDU. If an I-PDU is mapped statically to a Container it is always transported in the same position of the Container-PDU.
Rationale: Support scenarios of Container Mapping where no header shall be used to reduce bandwidth in case always all contained PDU shall be sent.
Use Case: Efficient transmission of small PDUs on high bandwidth busses
Dependencies: A Container-PDU shall either support static or dynamic mapping (see SRS_IpduM_02820) but no mixture.
–Last change on issue 76543 comment 100–

BW-C-Level:

Application	Specification	Bus
1	2	1