Document Title	TPS_SoftwareComponentTemplate: 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 TPS_SoftwareComponentTemplate

1.1 Specification Item constr_1004

Trace References:

none

Content:

The same In the scope of AtomicSwComponentType.AtomicSwComponentType.internalBehavior.InternalBehavior.dataTypeMapping, each ApplicationDataTypes may Type shall be mapped to different exactly one ImplementationDataTypes even in the scope of a single ECU (more exactly speaking, a single RTE), but not in the scope of a single atomic software componentType.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77040: Nested constraint in TPS_SWCT_01193

  Problem description:

  TPS_SWCT_01193 contains the definition of constr_1004, which is not allowed due to the existence of constr_2565 (STDT).
  –Last change on issue 77040 comment 3–

  Agreed solution:

  SWCT
  ====

  Remove constr_1004 and the note below from the list of bullet points inside TPS_SWCT_01193 and place constr_1004 and the note immediately below TPS_SWCT_01193.

  Replace the following sentence inside TPS_SWCT_01193

  Depending on the use case and on the scope, 1:n as well as n:1 mappings are possible:

  with

  Depending on the use case and on the scope, 1:n as well as n:1 mappings are possible (constr_1004 applies):

  TOOL

====

Let CheckDocumentSource complain about nested traces.
–Last change on issue 77040 comment 4–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.2 Specification Item constr_1007

Trace References:

none

Content:

The allowed attributes of SwDataDefProps for ApplicationDataTypes and their allowed multiplicities are listed as an overview in table [REF table_3a_CategoriesAppl]. 71082 7108276585

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #74212: Clarifications on DataConstrs w.r.t. optionallity and PhysConstrs versus InternalConstr

  Problem description:

  The figure Figure 5.6: Value ranges and invalid values shows very nice how the different data constraints needs to match to each other.

  1.) But the figure shows something like intern constrains of an ApplicationDataType. How des this match to TPS_SWCT_01288 which implies that for ApplicationDataTypes only PhysConstrs are applicable. Or is this a misinterpretation?

  2.) Since DataConstr are optional, I miss also some precise statement how the limits of a data type is defined in case of missing DataConstr.

  I would assume the following:
  If an ApplicationDataTypes does not define a DataConstr the limit of the later mapped ImplementationDataType applies.

If an ImplementationDataType does not define a DataConstr the technical limit of the BaseType defined by the size of type in bits and the according encoding applies.

Agreed solution:

1. Split the dataConstr line in tables 5.8, 5.18, 5.32, and 5.33 such that two new line are created

dataConstr.physConstr
dataConstr.internalConstr

Keep the existing setting for the respective flavor (application vs. implementation) and set the respective other line to to "d/c" and add an explanation to the legend that "d/c" stands for "don't care.

Update TPS_SWCT_01288:

[TPS_SWCT_01288] Interpretation of PhysConstrs and InternalConstrs by tools
DataConstr is an ARElement which can be reused by several data type specifications. Especially an ImplementationDataType and an Application-
DataType which are mapped to each other, can refer to the same constraints or they can define their own constraints.
To avoid conflicts, in both cases PhysConstrs shall be interpreted by tools only with respect to application data types while InternalConstrs shall be interpreted only with respect to implementation data types.
In case only physConstr are provided to ApplicationDataTypes the CompuMethod can be used to compute the internalConstr. ()

In Figure 5.6 (Value ranges and invalid values), replace the string "internalConstrs of ApplicationDataType" by "computed internalConstrs of ApplicationDataType"

2. Add note: If an ApplicationPrimitiveDataType does not define dataConstr then implicit constraints can be derived from physical meaning of the ApplicationDataType. For example, if the data type represents a temperature, the lower bound will not be able to exceed 0K. For other physical meanings it could be possible that the implicitly assumed limits go from -INF to +INF. In order to void ambiguity for the values of limits it is strongly recommended to define a reasonable limit for ApplicationPrimitiveDataTypes.
–Last change on issue 74212 comment 7–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.3 Specification Item constr_1009

Trace References:

none

Content:

A complete list of the SwDataDefProps and other attributes and their multiplicities which are allowed for a given Identifiable.category is shown in table [REF table_3a_Categories Impl]. 71082,73126, 76585

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #74212: Clarifications on DataConstrs w.r.t. optionallity and PhysConstrs versus InternalConstr

  Problem description:

  The figure Figure 5.6: Value ranges and invalid values shows very nice how the different data constraints needs to match to each other.

  1.) But the figure shows something like intern constrains of an ApplicationDataType. How des this match to TPS_SWCT_01288 which implies that for ApplicationDataTypes only PhysConstrs are applicable. Or is this a misinterpretation?

  2.) Since DataConstr are optional, I miss also some precise statement how the limits of a data type is defined in case of missing DataConstr.

  I would assume the following:
  If an ApplicationDataTypes does not define a DataConstr the limit of the later mapped ImplementationDataType applies.
  If an ImplementationDataType does not define a DataConstr the technical limit of the BaseType defined by the size of type in bits and the according encoding applies.

  Agreed solution:

  1. Split the dataConstr line in tables 5.8, 5.18, 5.32, and 5.33 such that two new line are created

  dataConstr.physConstr
  dataConstr.internalConstr

  Keep the existing setting for the respective flavor (application vs. implementation) and set the respective other line to to "d/c" and add an explanation to the legend that "d/c" stands for "don't care.

Update TPS_SWCT_01288:

[TPS_SWCT_01288] Interpretation of PhysConstrs and InternalConstrs by tools
DataConstr is an ARElement which can be reused by several data type specifications. Especially an ImplementationDataType and an ApplicationDataType which are mapped to each other, can refer to the same constraints or they can define their own constraints.
To avoid conflicts, in both cases PhysConstrs shall be interpreted by tools only with respect to application data types while InternalConstrs shall be interpreted only with respect to implementation data types.
In case only physConstr are provided to ApplicationDataTypes the CompuMethod can be used to compute the internalConstr. ()

In Figure 5.6 (Value ranges and invalid values), replace the string "internalConstrs of ApplicationDataType" by "computed internalConstrs of ApplicationDataType"

2. Add note: If an ApplicationPrimitiveDataType does not define dataConstr then implicit constraints can be derived from physical meaning of the ApplicationDataType. For example, if the data type represents a temperature, the lower bound will not be able to exceed 0K. For other physical meanings it could be possible that the implicitly assumed limits go from -INF to +INF. In order to void ambiguity for the values of limits it is strongly recommended to define a reasonable limit for ApplicationPrimitiveDataTypes.
–Last change on issue 74212 comment 7–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.4 Specification Item constr_1011

Trace References:

none

Content:

For the attribute SwBaseType.Identifiable.category only the values FIXED_LENGTHFIXED_LENGTH and VARIABLE_LENGTHVARIABLE_LENGTH, and VOIDVOID are supported.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #74446: Explanation for the applicability of VARIABLE_LENGTH missing

  Problem description:

  The SWCT contains constr_1011 and constr_1013 that mention the value VARIABLE_LENGTH as a possible setting for SwBaseType.category.

  There is no clear description when this value would be applicable. In fact, doubts are warranted whether the value (and with it, the attribute maxBaseTypeSize) should be applicable in AUTOSAR at all.

  Background: the definition of a SwBaseType is supposed to refer to an atomic data type. The "base type" for an array is still the "base type" of the array elements, as opposed to the definition of a base type for the array itself.

  The difference between the definition of a "base type" for the array elements and a "base type" for the entire array manifests in the attributes baseTypeSize, resp. maxBaseTypeSize.

  In some cases definitions of SwBaseType can be found in the wild that look similar to this specimen:

  <SW-BASE-TYPE>
  <SHORT-NAME>FunnyBaseType</SHORT-NAME>
  <CATEGORY>VARIABLE_LENGTH</CATEGORY>
  <MAX-BASE-TYPE-SIZE>1433</MAX-BASE-TYPE-SIZE>
  <BASE-TYPE-ENCODING>WINDOWS-1252</BASE-TYPE-ENCODING>
  </SW-BASE-TYPE>

  I guess that this base type has been build on the assumption that a string deserves a base type for the entire Windows-1252-encoded string.

  This assumption would not only result in countless definitions of SwBaseTypes for each possible combination of array length and data type, it would also create an unholy dependency to the value of ImplementationDataTypeElement.arraySize. In other words, if the value of array size changes than the value of maxBaseTypeSize would have to change accordingly.

  This does not make much sense.

  I tried to get some information about the use case for VARIABLE_LENGTH that was originally intended. Bernhard mentioned that it was meant to support edge cases for the generation of A2L files, in particular for string implementations where the first element denotes the length and the rest is reserved for the payload.

If there are further known use cases, please feel free to leave a comment ...

I would like to provide a better explanation to the applicability of VARIABLE_LENGTH and also better explain that the SwBaseType is not meant to be applied to composite data types.

At the moment the document is rather indifferent about these two aspects.

Agreed solution:

SWCT:
—–

1. Reformulate:

[constr_1011] category of SwBaseType
d The attribute SwBaseType.category must be set and only the values FIXED_LENGTH and VOID are supported. c

[constr_1422] Value of category is VOID
d If the value of the attribute SwBaseType.category is set to VOID then the attribute baseTypeSize shall not exist. c()

[constr_1012] Value of category is FIXED_LENGTH
d If the value of the attribute SwBaseType.category is set to FIXED_LENGTH then the attribute baseTypeSize shall be filled with content. c()

[constr_1229] category of ImplementationDataType boils down to VALUE
subElement.category is set to VALUE or TYPE_REFERENCE that eventually boils down to VALUE and the subElement refers to a SwBaseType where baseTypeSize is set to the value 8 and the baseTypeEncoding is set to NONE.

[constr_1220] Compatibility of SwBaseType
d Two SwBaseTypes are compatible if and only if attributes baseTypeSize respectively byteOrder, memAlignment, baseTypeEncoding, and nativeDeclaration have identical values. c()

2. Remove [constr_1013].

3. Set attribute maxBaseTypeSize to 'obsolete'.

4. Adapt Listing 5.4: Example for the definition of a string SwBaseType: (add CATEGORY)

```
<AR-PACKAGE>
<ELEMENTS>
<SW-BASE-TYPE>
<SHORT-NAME>MyTextBaseType</SHORT-NAME>
<CATEGORY>FIXED_LENGTH</CATEGORY>
<BASE-TYPE-SIZE>8</BASE-TYPE-SIZE>
<BASE-TYPE-ENCODING>UTF-8</BASE-TYPE-ENCODING>
</SW-BASE-TYPE>
<SW-BASE-TYPE>
<SHORT-NAME>uint8BaseType</SHORT-NAME>
<CATEGORY>FIXED_LENGTH</CATEGORY>
<BASE-TYPE-SIZE>8</BASE-TYPE-SIZE>
</SW-BASE-TYPE>
</ELEMENTS>
</AR-PACKAGE>
```

5. Fix all object diagrams showing SwBaseType (add CATEGORY).

DEXT:
—–

1. Remove in Table 4.8: Allowed attributes of SwDataDefProps for DiagnosticDataElement.swDataDefProps the line baseType.baseTypeDefinition.maxBaseTypeSize N/A.
–Last change on issue 74446 comment 7–

BW-C-Level:

Application	Specification	Bus
3	3	1

1.5 Specification Item constr_1012

Trace References:

none

Content:

If the value of the attribute SwBaseType.Identifiable.category is set to FIXED_LENGTH then the attribute BaseTypeDirectDefinition.baseTypeSize shall be filled with contentand attribute BaseTypeDirectDefinition.maxBaseTypeSize shall not exist.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #74446: Explanation for the applicability of VARIABLE_LENGTH missing

  Problem description:

  The SWCT contains constr_1011 and constr_1013 that mention the value VARIABLE_LENGTH as a possible setting for SwBaseType.category.

  There is no clear description when this value would be applicable. In fact, doubts are warranted whether the value (and with it, the attribute maxBaseTypeSize) should be applicable in AUTOSAR at all.

  Background: the definition of a SwBaseType is supposed to refer to an atomic data type. The "base type" for an array is still the "base type" of the array elements, as opposed to the definition of a base type for the array itself.

  The difference between the definition of a "base type" for the array elements and a "base type" for the entire array manifests in the attributes baseTypeSize, resp. maxBaseTypeSize.

  In some cases definitions of SwBaseType can be found in the wild that look similar to this specimen:

  ```
  <SW-BASE-TYPE>
  <SHORT-NAME>FunnyBaseType</SHORT-NAME>
  <CATEGORY>VARIABLE_LENGTH</CATEGORY>
  <MAX-BASE-TYPE-SIZE>1433</MAX-BASE-TYPE-SIZE>
  <BASE-TYPE-ENCODING>WINDOWS-1252</BASE-TYPE-ENCODING>
  </SW-BASE-TYPE>
  ```

  I guess that this base type has been build on the assumption that a string deserves a base type for the entire Windows-1252-encoded string.

  This assumption would not only result in countless definitions of SwBaseTypes for each possible combination of array length and data type, it would also create an unholy dependency to the value of ImplementationDataTypeElement.arraySize. In other words, if the value of array size changes than the value of maxBaseTypeSize would have to change accordingly.

This does not make much sense.

I tried to get some information about the use case for VARIABLE_LENGTH that was originally intended. Bernhard mentioned that it was meant to support edge cases for the generation of A2L files, in particular for string implementations where the first element denotes the length and the rest is reserved for the payload.

If there are further known use cases, please feel free to leave a comment ...

I would like to provide a better explanation to the applicability of VARIABLE_LENGTH and also better explain that the SwBaseType is not meant to be applied to composite data types.

At the moment the document is rather indifferent about these two aspects.

Agreed solution:

SWCT:
—–

1. Reformulate:

[constr_1011] category of SwBaseType
d The attribute SwBaseType.category must be set and only the values FIXED_LENGTH and VOID are supported. c

[constr_1422] Value of category is VOID
d If the value of the attribute SwBaseType.category is set to VOID then the attribute baseTypeSize shall not exist. c()

[constr_1012] Value of category is FIXED_LENGTH
d If the value of the attribute SwBaseType.category is set to FIXED_LENGTH then the attribute baseTypeSize shall be filled with content. c()

[constr_1229] category of ImplementationDataType boils down to VALUE
subElement.category is set to VALUE or TYPE_REFERENCE that eventually boils down to VALUE and the subElement refers to a SwBaseType where baseTypeSize is set to the value 8 and the baseTypeEncoding is set to NONE.

[constr_1220] Compatibility of SwBaseType
d Two SwBaseTypes are compatible if and only if attributes baseTypeSize respectively byteOrder, memAlignment, baseTypeEncoding, and nativeDeclaration have identical values. c()

2. Remove [constr_1013].

3. Set attribute maxBaseTypeSize to ’obsolete’.

4. Adapt Listing 5.4: Example for the definition of a string SwBaseType: (add CATEGORY)

```
<AR-PACKAGE>
<ELEMENTS>
<SW-BASE-TYPE>
<SHORT-NAME>MyTextBaseType</SHORT-NAME>
<CATEGORY>FIXED_LENGTH</CATEGORY>
<BASE-TYPE-SIZE>8</BASE-TYPE-SIZE>
<BASE-TYPE-ENCODING>UTF-8</BASE-TYPE-ENCODING>
</SW-BASE-TYPE>
<SW-BASE-TYPE>
<SHORT-NAME>uint8BaseType</SHORT-NAME>
<CATEGORY>FIXED_LENGTH</CATEGORY>
<BASE-TYPE-SIZE>8</BASE-TYPE-SIZE>
</SW-BASE-TYPE>
</ELEMENTS>
</AR-PACKAGE>
```

5. Fix all object diagrams showing SwBaseType (add CATEGORY).

DEXT:
—–

1. Remove in Table 4.8: Allowed attributes of SwDataDefProps for DiagnosticDataElement.swDataDefProps the line baseType.baseTypeDefinition.maxBaseTypeSize N/A.
–Last change on issue 74446 comment 7–

BW-C-Level:

Application	Specification	Bus
3	3	1

1.6 Specification Item constr_1013

Trace References:

none

Content:

If the value of the attribute SwBaseType.Identifiable.category is set to VARIABLE_LENGTH then the attribute BaseTypeDirectDefinition.maxBaseTypeSize shall be filled with content and attribute BaseTypeDirectDefinition.baseTypeSize shall not exist.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #74446: Explanation for the applicability of VARIABLE_LENGTH missing

  Problem description:

  The SWCT contains constr_1011 and constr_1013 that mention the value VARIABLE_LENGTH as a possible setting for SwBaseType.category.

  There is no clear description when this value would be applicable. In fact, doubts are warranted whether the value (and with it, the attribute maxBaseTypeSize) should be applicable in AUTOSAR at all.

  Background: the definition of a SwBaseType is supposed to refer to an atomic data type. The "base type" for an array is still the "base type" of the array elements, as opposed to the definition of a base type for the array itself.

  The difference between the definition of a "base type" for the array elements and a "base type" for the entire array manifests in the attributes baseTypeSize, resp. maxBaseTypeSize.

  In some cases definitions of SwBaseType can be found in the wild that look similar to this specimen:

  ```
  <SW-BASE-TYPE>
  <SHORT-NAME>FunnyBaseType</SHORT-NAME>
  <CATEGORY>VARIABLE_LENGTH</CATEGORY>
  <MAX-BASE-TYPE-SIZE>1433</MAX-BASE-TYPE-SIZE>
  <BASE-TYPE-ENCODING>WINDOWS-1252</BASE-TYPE-ENCODING>
  </SW-BASE-TYPE>
  ```

  I guess that this base type has been build on the assumption that a string deserves a base type for the entire Windows-1252-encoded string.

  This assumption would not only result in countless definitions of SwBaseTypes for each possible combination of array length and data type, it would also create an unholy dependency to the value of ImplementationDataTypeElement.arraySize. In other words, if the value of array size changes than the value of maxBaseTypeSize would have to change accordingly.

This does not make much sense.

I tried to get some information about the use case for VARIABLE_LENGTH that was originally intended. Bernhard mentioned that it was meant to support edge cases for the generation of A2L files, in particular for string implementations where the first element denotes the length and the rest is reserved for the payload.

If there are further known use cases, please feel free to leave a comment ...

I would like to provide a better explanation to the applicability of VARIABLE_LENGTH and also better explain that the SwBaseType is not meant to be applied to composite data types.

At the moment the document is rather indifferent about these two aspects.

Agreed solution:

SWCT:
—–

1. Reformulate:

[constr_1011] category of SwBaseType
d The attribute SwBaseType.category must be set and only the values FIXED_LENGTH and VOID are supported. c

[constr_1422] Value of category is VOID
d If the value of the attribute SwBaseType.category is set to VOID then the attribute baseTypeSize shall not exist. c()

[constr_1012] Value of category is FIXED_LENGTH
d If the value of the attribute SwBaseType.category is set to FIXED_LENGTH then the attribute baseTypeSize shall be filled with content. c()

[constr_1229] category of ImplementationDataType boils down to VALUE
subElement.category is set to VALUE or TYPE_REFERENCE that eventually boils down to VALUE and the subElement refers to a SwBaseType where baseTypeSize is set to the value 8 and the baseTypeEncoding is set to NONE.

[constr_1220] Compatibility of SwBaseType d Two SwBaseTypes are compatible if and only if attributes baseTypeSize respectively byteOrder, memAlignment, baseTypeEncoding, and nativeDeclaration have identical values. c()

2. Remove [constr_1013].

3. Set attribute maxBaseTypeSize to ’obsolete’.

4. Adapt Listing 5.4: Example for the definition of a string SwBaseType: (add CATEGORY)

```
<AR-PACKAGE>
<ELEMENTS>
<SW-BASE-TYPE>
<SHORT-NAME>MyTextBaseType</SHORT-NAME>
<CATEGORY>FIXED_LENGTH</CATEGORY>
<BASE-TYPE-SIZE>8</BASE-TYPE-SIZE>
<BASE-TYPE-ENCODING>UTF-8</BASE-TYPE-ENCODING>
</SW-BASE-TYPE>
<SW-BASE-TYPE>
<SHORT-NAME>uint8BaseType</SHORT-NAME>
<CATEGORY>FIXED_LENGTH</CATEGORY>
<BASE-TYPE-SIZE>8</BASE-TYPE-SIZE>
</SW-BASE-TYPE>
</ELEMENTS>
</AR-PACKAGE>
```

5. Fix all object diagrams showing SwBaseType (add CATEGORY).

DEXT:
—–

1. Remove in Table 4.8: Allowed attributes of SwDataDefProps for DiagnosticDataElement.swDataDefProps the line baseType.baseTypeDefinition.maxBaseTypeSize N/A.
–Last change on issue 74446 comment 7–

BW-C-Level:

Application	Specification	Bus
3	3	1

1.7 Specification Item constr_1015

Trace References:

none

Content:

The prioritization and usage of attributes of meta-class SwDataDefProps shall follow the restrictions given in table [REF table_3a_DataDefPropsUsageDetails]. 72035,72418,78349

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #78017: Specification gap how to process VariationPointProxy with swSystemconst

  Problem description:

  Name: Thomas Ziegler
  Phone: 071181146621
  Role: Software Developer

  Description/Motivation:

  For the usage of a swSystemconst in C-Code, it is required to define a VariationPointProxy. The AUTOSAR element swSystemconst may reference a CompuMethod. Unfortunately there is a gap how to process the CompuMethod of a swSystemconst, if this swSystemconst is referenced by VariationPointProxy.
  In the application layer the physics is specified. If a swSystemconst is converted from application level to implementation level, in this case the value should be converted to the implementation value, too. This should be done by the CompuMethod of the swSystemconst.

  Use case for a physical swSystemconst:
  A function developer defines a swSystemconst EpmMax which represents the maximal value of the engine speed to which a CompuMethod CM_EpmMax of category LINEAR (e.g. with a formula int=2*phys) is attached. Finally, the internal value of the swSystemconst shall be read/accessed in the C-Code implementations.

  Was there already a decision? No

Agreed solution:

GST:
—-

1. Reformulate the attribute swDataDefProps for SwSystemconst:
This denotes the data definition properties of the system constant. This supports to express the limits and optionally a conversion within the internal to physical values by a compu method. -> assign to GST

2. Remove "For more details please refer to [20]." below [TPS_GST_00262].

3.

"Such a system constant is basically a name/value pair. shortName, dataConstr and compuMethod
for a system constant are defined in SwSystemconst."

Extend the paragraph by : Similar to data types an unit can be explicitly expressed by the unit reference.

and insert below the following text block:

DataConstr are defined to describe limits to the applicable SwSystemconstValue(s). The compuMethod supports either the conversion between internal values and textual literals or the conversion between internal values and the physical meaning. Regardless of the existence of a compuMethod the evaluation of formula containing SwSystemconst is realized by internal values. Therefore also the SwSystemconstValue(s) hold the internal values. Nevertheless the compuMethod can be used by tools to show the physical value.

RTE:
—-

Insert below:

This requirements makes the SwSystemconst values available to resolve the prebuild
variability in the software components via the Preprocessor. This might
be used to
- read the actual value of the value assigned to a SwSystemconst
- read the setting of an attribute (e.g. array size) dependent from a SwSystemconst
- check the existence of a conditional existent object, e.g. an code fragment

implementing a particular functionality

a new paragraph:

Please note the Rte_SysCon macro holds the internal value of the evaluated the AttributeValueVariationPoint or ConditionByFormula. Therefore the RTE does not perform value conversions for SwSystemconst using a compuMethod. See [TPS_GST_00262].

SWCT:
—–

Allow the unit for Systemconst in Table 5.41: Usage of Attributes of SwDataDefProps. Add a "D" at the intersection of unit and Systemconst.
–Last change on issue 78017 comment 4–

BW-C-Level:

Application	**Specification**	**Bus**
1	1	1

1.8 Specification Item constr_1038

Trace References:

none

Content:

A ClientServerOperation.possibleError referenced by a ClientServerOperation shall be owned by the ~~ClientServer~~PortInterface that also owns the ClientServerOperation.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76526: TPS_SWCT_01490 is no longer true, thanks to the adaptive platform

  Problem description:

  Because ApplicationErrors are also used on the adaptive platform, TPS_SWCT_01490 and constr_1038 are no longer true in its literal meaning.

  Either we remove the specification item or else we should add some further qualification such that the statement is explicitly limited to the classic platform.
  –Last change on issue 76526 comment 3–

Agreed solution:

Remove TPS_SWCT_01490

Update the existing phrasing of const_1038

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

to

[constr_1038] Reference to ApplicationError
A possibleError referenced by a ClientServerOperation shall be owned by the PortInterface that also owns the ClientServerOperation.
–Last change on issue 76526 comment 4–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.9 Specification Item constr_1044

Trace References:

none

Content:

According to the origin of DataFilter, i.e. OSEK COM 3.0.3 specification OSEKISO 17356-4 specification ISO_2d_COM17356_2d_4, DataFilters can only be applied to values with an integer base type.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73564: References to OSEK

  Problem description:

  The OSEK web site is not available.
  References need to be updated.

  OSEK was pushed to ISO 17356
  ISO 17356-2: introduction?

ISO 17356-3: OS
ISO 17356-4: COM
ISO 17356-5: NM (not sure we need to have such reference)
ISO 17356-6: OIL (not sure we need to have such reference)

I tried to select the documents where it would make sense to get a fix.

Agreed solution:

Replace references to
* OSEK web site
* the OSEK file name / version
with an ISO reference (with ISO version)

Check references to OSEK subsection (or avoid such references), to make sure that the section numbering in ISO is the same.

SWS OS
======

Change references in chapter 3.2.1 to ISO. Remove [16], [18], [19] and [20]

Remove [22] from chapter 3.2.2

Remove "OSEKtime OS [16] and the HIS Protected OSEK [22] are immature specifications that contain concepts necessary for AUTOSAR and satisfy specific application domains. It is the purpose of this document to identify these needs and to recommend the use of parts (or all) of these specifications as appropriate." from chapter 4.1.

Change "OSEK OS" to new ISO reference ("OSEK OS[..]" - many occurences, also in chapter 10)

Remove "So called hard and smooth synchronization from OSEKtime [16] are supported by this single unified concept in AUTOSAR OS. Smooth synchronization may be emulated by setting the small adjustment values on the final expiry point. Hard synchronization may be emulated by setting large adjustment values on the final expiry point." from 7.4.2.2.3

Remove "and provides the type of protection given by the OSEKtime Interrupt re-enable
schedule event [16]." from 7.7.2.1

Remove chapter 12.3

SWCT:
=====

Replace reference to OSEK COM:

Fifteen filter algorithms formally described by the enumeration type DataFilterTypeEnum in the meta-model are taken from OSEK COM 3.0.3 specification [18] that is referenced by the RTE specification [2].

by:

Fifteen filter algorithms formally described by the enumeration type DataFilterTypeEnum in the meta-model are taken from the ISO 17356-4 specification [18] that is referenced by the RTE specification [2].

Replace:

[TPS_SWCT_01222] Applicability of DataFilter
This OSEK specification states that filtering is only used for messages that can be interpreted as C language unsigned integer types (characters, unsigned integers and enumerations).(RS_SWCT_03221)

by:

[TPS_SWCT_01222] Applicability of DataFilter
The ISO 17356-4 specification states that filtering is only used for messages that can be interpreted as C language unsigned integer types (characters, unsigned integers and enumerations).(RS_SWCT_03221)

Replace:

[constr_1044] Applicability of DataFilter
According to the origin of DataFilter, i.e. OSEK COM 3.0.3 specification [18], DataFilters can only be applied to values with an integer base type.()

by:

[constr_1044] Applicability of DataFilter
According to the origin of DataFilter, i.e. ISO 17356-4 specification [18], DataFilters can only be applied to values with an integer base type.()

Replace footnote to [constr_1090]:

This constraint is valid at least in the OSEK standard where an extended task (that can have wait points) can only exist a single time in the context of the scheduler.

by:

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

============
Dem
============
Replace the reference [17] Communication in Chapter 3.1 Input documents & related standards and norms Bibliography by : ISO 17356-3 in www.iso.org/

============
Dcm
============
Replace the reference [8]Communication in Chapter 3.1 Input documents & related standards and norms Bibliography by : ISO 17356-3 in www.iso.org/

============
EXP_VFB

Add an entry in chapter "13 References" related to ISO 17356-4:
ISO 17356-4
OSEK/VDX Communication (COM)
www.iso.org

Change the references to "OSEK-COM V3.0.3" in Table 4.2 (4.3.2 From the point of view of the receiver) and EXP_Vfb_00028 (4.3.4 Filtering between the sender and the receiver) to "ISO 17356-4". Link the references to the entry in chapter "13 References".

============
SRS_BSWGeneral

5 General Requirements on Basic Software

Replace "OSEK OS" by "ISO 17356-3"

5.2.3.4 Standard header Files
Replace in [SRS_BSW_00348]

Because E_OK is already defined within OSEK OS, E_OK has to be checked for being already defined:
/* for OSEK compliance this typedef has been added */

by

Because E_OK is already defined within ISO 17356-3, E_OK has to be checked for being already defined
/* for ISO 17356 compliance this typedef has been added */

6.2 Related Standards and Norms

Remove
6.2.1 OSEK
[STD_OSEK_OS] OSEK/VDX Operating System Specification
http://www.osek-vdx.org

Replace with ISO 17356-3 norm

============
SRS_COM

1)
Remove [DOC_OSEK_GLOS] and all its references, since a) AUTOSAR has an own wording and glossary b) the references to (old) OSEK-terms most probably create more confusion than guidance

2)
Set reference of [DOC_OSEK_COM] to: ISO 17356-4: COM
Move refernce to Chapter 7.3 ISO and remove Chapter 7.2 OSEK

3)
Remove secton references to [DOC_OSEK_COM], just keep the textual references no numbers

4)
Search and replace textual references to OSEK COM 3.0.3 by ISO 17356-4: COM

5)

[SRS_Com_02084]: Change following text in Description,
from
< The possibilities to define those conditions shall be the same as defined in [DOC_OSEK_COM] reception filter algorithms (see [DOC_OSEK_GLOS], Section 2.2.2).
to
> The possibilities to define those conditions shall be the same as defined in [DOC_ISO_COM] reception filter algorithms (see [DOC_ISO_COM], Section 3.2.3).

6)
[SRS_Com_02058]: Change following text in Supporting Material,
from
< If no update bits are used, the AUTOSAR COM module provides the deadline monitoring defined in [DOC_OSEK_COM] (Section 2.5.1).
to
> If no update bits are used, the AUTOSAR COM module provides the deadline monitoring defined in [DOC_ISO_COM] (Section 3.5.1).

============
SRS_Os

in chapter 6.2.1 OSEK:

Replace "[STD_OSEK_OS] OSEK/VDX Operating System, Version 2.2.3, http://www.osek-vdx.org/mirror/os223.pdf"
by "[STD_OSEK_OS] ISO 17356-3: OS"

Replace "[STD_OSEK_OIL] OSEK / VDX Implementation Language (OIL) V2.5, OSEK Implementation Language, http://www.osek-vdx.org/mirror/oil25.pdf"
by "[STD_OSEK_OIL] ISO 17356-6: OIL"

Remove "[STD_OSEK_TTOS] OSEK/VDX Time-Triggered Operating System, Version 1.0, July 24, 2001, http://www.osek-vdx.org/mirror/ttos10.pdf"

Remove "[STD_OSEK_ORTI] OSEK/VDX ORTI (OSEK RunTime Interface) Part A Version 2.1.1, Part B Version 2.1, http://www.osek-vdx.org/mirror/ORTI-A-211.pdf"

in [SRS_Os_11002]: remove [STD_OSEK_TTOS] from Supporting Material

============
SWS_COM

1)
Set reference of [17] to ISO 17356-4: COM
Set reference of [18] to ISO 17356-6: OIL

2)
Search and replace textual references to OSEK COM 3.0.3 by ISO 17356-4: COM or [17] (if suitable)
and textual references to OSEK OIL by ISO 17356-6: OIL or [18] (if suitable)

============
SWS_StandardTypes:
In Section 3.2: replace
[7] OSEK/VDX Operating System, Version 2.2.2 www.osek-vdx.org/os222.pdf
by
[7] OSEK/VDX Operating System, ISO 17356-3: OS
============
CP_TR_AutosarModelConstraints
see PS for the SWCT.
============

==
SRS_NetworkManagement:

1) Change Section 7.2.1 name from "OSEK" to "ISO 17356-5"
Remove the description:
[5] [STD_OSEK_NM]
OSEK/VDX NM Specification (ISO 17356-5), Version 2.5.3
[STD_OSEK_NM] OSEK/VDX NM Specification (ISO 17356-5), V2.5.3
http://www.osek-vdx.org/
Change to :
OSEK/VDX NM Specification
www.iso.org
===>
7.2.1 ISO 17356-5
[5] ISO 17356-5: NM Specification
www.iso.org

2) Replace "OSEK-NM" to "ISO 17356-5: NM Specification" in [SRS_Nm_02515]

3) Replace "OSEK NM 2.5.3" to "ISO 17356-5: NM Specification" in [SRS_Nm_00142]
–Last change on issue 73564 comment 28–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.10 Specification Item constr_1090

Trace References:

none

Content:

A single RunnableEntity can actually wait only at a single WaitPoint provided that the RunnableEntity can only be scheduled a single time This constraint is valid at least in the OSEK ISO 17356-3 ISO_2d_17356_2d_3 standard where an extended task (that can have wait points) can only exist a single time in the context of the scheduler. .

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73564: References to OSEK

  Problem description:

  The OSEK web site is not available.
  References need to be updated.

  OSEK was pushed to ISO 17356
  ISO 17356-2: introduction?
  ISO 17356-3: OS
  ISO 17356-4: COM
  ISO 17356-5: NM (not sure we need to have such reference)
  ISO 17356-6: OIL (not sure we need to have such reference)

  I tried to select the documents where it would make sense to get a fix.

  Agreed solution:

  Replace references to
  * OSEK web site
  * the OSEK file name / version
  with an ISO reference (with ISO version)

  Check references to OSEK subsection (or avoid such references), to make sure that the section numbering in ISO is the same.

SWS OS
======

Change references in chapter 3.2.1 to ISO. Remove [16], [18], [19] and [20]

Remove [22] from chapter 3.2.2

Remove "OSEKtime OS [16] and the HIS Protected OSEK [22] are immature specifications that contain concepts necessary for AUTOSAR and satisfy specific application domains. It is the purpose of this document to identify these needs and to recommend the use of parts (or all) of these specifications as appropriate." from chapter 4.1.

Change "OSEK OS" to new ISO reference ("OSEK OS[..]" - many occurences, also in chapter 10)

Remove "So called hard and smooth synchronization from OSEKtime [16] are supported by this single unified concept in AUTOSAR OS. Smooth synchronization may be emulated by setting the small adjustment values on the final expiry point. Hard synchronization may be emulated by setting large adjustment values on the final expiry point." from 7.4.2.2.3

Remove "and provides the type of protection given by the OSEKtime Interrupt re-enable
schedule event [16]." from 7.7.2.1

Remove chapter 12.3

SWCT:
=====

Replace reference to OSEK COM:

Fifteen filter algorithms formally described by the enumeration type DataFilterTypeEnum in the meta-model are taken from OSEK COM 3.0.3 specification [18] that is referenced by the RTE specification [2].

by:

Fifteen filter algorithms formally described by the enumeration type DataFilter-

TypeEnum in the meta-model are taken from the ISO 17356-4 specification [18] that is referenced by the RTE specification [2].

Replace:

[TPS_SWCT_01222] Applicability of DataFilter
This OSEK specification states that filtering is only used for messages that can be interpreted as C language unsigned integer types (characters, unsigned integers and enumerations).(RS_SWCT_03221)

by:

[TPS_SWCT_01222] Applicability of DataFilter
The ISO 17356-4 specification states that filtering is only used for messages that can be interpreted as C language unsigned integer types (characters, unsigned integers and enumerations).(RS_SWCT_03221)

Replace:

[constr_1044] Applicability of DataFilter
According to the origin of DataFilter, i.e. OSEK COM 3.0.3 specification [18], DataFilters can only be applied to values with an integer base type.()

by:

[constr_1044] Applicability of DataFilter
According to the origin of DataFilter, i.e. ISO 17356-4 specification [18], DataFilters can only be applied to values with an integer base type.()

Replace footnote to [constr_1090]:

This constraint is valid at least in the OSEK standard where an extended task (that can have wait points) can only exist a single time in the context of the scheduler.

by:

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

============
Dem
============
Replace the reference [17] Communication in Chapter 3.1 Input documents &

related standards and norms Bibliography by : ISO 17356-3 in www.iso.org/

============
Dcm
============
Replace the reference [8]Communication in Chapter 3.1 Input documents & related standards and norms Bibliography by : ISO 17356-3 in www.iso.org/

============
EXP_VFB

Add an entry in chapter "13 References" related to ISO 17356-4:
ISO 17356-4
OSEK/VDX Communication (COM)
www.iso.org

Change the references to "OSEK-COM V3.0.3" in Table 4.2 (4.3.2 From the point of view of the receiver) and EXP_Vfb_00028 (4.3.4 Filtering between the sender and the receiver) to "ISO 17356-4". Link the references to the entry in chapter "13 References".

============
SRS_BSWGeneral

5 General Requirements on Basic Software

Replace "OSEK OS" by "ISO 17356-3"

5.2.3.4 Standard header Files
Replace in [SRS_BSW_00348]

Because E_OK is already defined within OSEK OS, E_OK has to be checked for being already defined:
/* for OSEK compliance this typedef has been added */

by

Because E_OK is already defined within ISO 17356-3, E_OK has to be checked for being already defined
/* for ISO 17356 compliance this typedef has been added */

6.2 Related Standards and Norms

Remove
6.2.1 OSEK
[STD_OSEK_OS] OSEK/VDX Operating System Specification
http://www.osek-vdx.org

Replace with ISO 17356-3 norm

============
SRS_COM

1)
Remove [DOC_OSEK_GLOS] and all its references, since a) AUTOSAR has an own wording and glossary b) the references to (old) OSEK-terms most probably create more confusion than guidance

2)
Set reference of [DOC_OSEK_COM] to: ISO 17356-4: COM
Move refernce to Chapter 7.3 ISO and remove Chapter 7.2 OSEK

3)
Remove secton references to [DOC_OSEK_COM], just keep the textual references no numbers

4)
Search and replace textual references to OSEK COM 3.0.3 by ISO 17356-4: COM

5)
[SRS_Com_02084]: Change following text in Description,
from
< The possibilities to define those conditions shall be the same as defined in [DOC_OSEK_COM] reception filter algorithms (see [DOC_OSEK_GLOS], Section 2.2.2).
to
> The possibilities to define those conditions shall be the same as defined in [DOC_ISO_COM] reception filter algorithms (see [DOC_ISO_COM], Section 3.2.3).

6)
[SRS_Com_02058]: Change following text in Supporting Material,
from
< If no update bits are used, the AUTOSAR COM module provides the deadline monitoring defined in [DOC_OSEK_COM] (Section 2.5.1).
to

> If no update bits are used, the AUTOSAR COM module provides the deadline monitoring defined in [DOC_ISO_COM] (Section 3.5.1).

============
SRS_Os

in chapter 6.2.1 OSEK:

Replace "[STD_OSEK_OS] OSEK/VDX Operating System, Version 2.2.3, http://www.osek-vdx.org/mirror/os223.pdf"
by "[STD_OSEK_OS] ISO 17356-3: OS"

Replace "[STD_OSEK_OIL] OSEK / VDX Implementation Language (OIL) V2.5, OSEK Implementation Language, http://www.osek-vdx.org/mirror/oil25.pdf"
by "[STD_OSEK_OIL] ISO 17356-6: OIL"

Remove "[STD_OSEK_TTOS] OSEK/VDX Time-Triggered Operating System, Version 1.0, July 24, 2001, http://www.osek-vdx.org/mirror/ttos10.pdf"

Remove "[STD_OSEK_ORTI] OSEK/VDX ORTI (OSEK RunTime Interface) Part A Version 2.1.1, Part B Version 2.1, http://www.osek-vdx.org/mirror/ORTI-A-211.pdf"

in [SRS_Os_11002]: remove [STD_OSEK_TTOS] from Supporting Material

============
SWS_COM

1)
Set reference of [17] to ISO 17356-4: COM
Set reference of [18] to ISO 17356-6: OIL

2)
Search and replace textual references to OSEK COM 3.0.3 by ISO 17356-4: COM or [17] (if suitable)
and textual references to OSEK OIL by ISO 17356-6: OIL or [18] (if suitable)

============
SWS_StandardTypes:
In Section 3.2: replace
[7] OSEK/VDX Operating System, Version 2.2.2 www.osek-vdx.org/os222.pdf
by
[7] OSEK/VDX Operating System, ISO 17356-3: OS

============
CP_TR_AutosarModelConstraints
see PS for the SWCT.
============

===
SRS_NetworkManagement:

1) Change Section 7.2.1 name from "OSEK" to "ISO 17356-5"
Remove the description:
[5] [STD_OSEK_NM]
OSEK/VDX NM Specification (ISO 17356-5), Version 2.5.3
[STD_OSEK_NM] OSEK/VDX NM Specification (ISO 17356-5), V2.5.3
http://www.osek-vdx.org/
Change to :
OSEK/VDX NM Specification
www.iso.org
===>
7.2.1 ISO 17356-5
[5] ISO 17356-5: NM Specification
www.iso.org

2) Replace "OSEK-NM" to "ISO 17356-5: NM Specification" in [SRS_Nm_02515]

3) Replace "OSEK NM 2.5.3" to "ISO 17356-5: NM Specification" in [SRS_Nm_00142]
–Last change on issue 73564 comment 28–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.11 Specification Item constr_1102

Trace References:

none

Content:

A If a SwComponentType may have has PortPrototypes typed by different PortClient ServerInterfaces with equal Referrable.shortName but conflicting and ApplicationErrors
.

defined then the following condition applies: ApplicationErrors are considered conflicting if ApplicationErrors with the same Referrable.shortName do have different shall have identical values of ApplicationError.errorCodes.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76525: Convert constr_1102 to specification item

  Problem description:

  The statement made by constr_1102 is pretty atypical for a constraint, i.e. there is nothing to check and report in case of validation.

  This has the character of a specification item and therefore it shall be made one.

  Agreed solution:

  Reformulate

  [constr_1102] ApplicationError in the scope of one SwComponentType d A
  SwComponentType may have PortPrototypes typed by different PortInterfaces with equal shortName but conflicting ApplicationErrors.
  ApplicationErrors are considered conflicting if ApplicationErrors with the same shortName do have different errorCodes.

  to

  [constr_1102] ApplicationError in the scope of one SwComponentType d
  If a SwComponentType has PortPrototypes typed by different ClientServerInterfaces with equal shortName and ApplicationErrors defined then the following condition applies: ApplicationErrors with the same shortName shall have identical values of errorCodes.

  Rationale: the RTE generator creates symbols for the error codes in which the shortName of the ClientServerInterface and the shortName of the ApplicationError occur.
  –Last change on issue 76525 comment 1–

  BW-C-Level:

Application	Specification	Bus
1	1	1

1.12 Specification Item constr_1126

Trace References:

none

Content:

The DataConstr (e.g. the limits) defined by the type of the providing data element shall be within the constraints defined by the type of the requiring data element.

For client-server communication, the following rules apply:

- For ClientServerOperation.arguments with attribute ArgumentDataPrototype.direction set to the value ArgumentDirectionEnum.in, the client shall take the role of the provider and the server shall take the role of the requiring side.
- For ClientServerOperation.arguments with attribute ArgumentDataPrototype.direction set to the value ArgumentDirectionEnum.inoutthe DataConstr shall be equal on both sides.
- For ClientServerOperation.arguments with attribute ArgumentDataPrototype.direction set to the value ArgumentDirectionEnum.out, the server shall take the role of the provider and the client shall take the role of the requiring side.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #75169: Compatibility of DataConstr in the context of C/S communication

  Problem description:

  I'm referring to the DataConstr compatibility rule constr_1126.
  My question is how to understand this constraint if the DataConstr are validated in the context of an ArgumentDataPrototype between a client and a server? I think it is not sufficient to see the client as a requester and the server as the provider. In my opinion the following cases must be distinguished (proposal):
  - for IN arguments, the client must be seen as the provider and the server as the requester
  - for INOUT arguments the DataConstr must be equal
  - for OUT arguments the server must be seen as the provider and the client as the requester

  Could we clarify this in the SWCT?

Agreed solution:

Add the following to constr_1126:

For client-server communication, the following rules apply:

- for IN arguments, the client shall be seen as the provider and the server as the requester
- for INOUT arguments the DataConstr shall be equal
- for OUT arguments the server shall be seen as the provider and the client as the requester
–Last change on issue 75169 comment 11–

BW-C-Level:

Application	Specification	Bus
4	4	1

1.13 Specification Item constr_1220

Trace References:

none

Content:

Two SwBaseTypes are compatible if and only if attributes BaseTypeDirectDefinition.baseTypeSize respectively BaseTypeDirectDefinition.maxBaseTypeSize, BaseTypeDirectDefinition.byteOrder, BaseTypeDirectDefinition.memAlignment, BaseTypeDirectDefinition.baseTypeEncoding, and BaseTypeDirectDefinition.nativeDeclaration have identical values.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #74446: Explanation for the applicability of VARIABLE_LENGTH missing

  Problem description:

  The SWCT contains constr_1011 and constr_1013 that mention the value VARIABLE_LENGTH as a possible setting for SwBaseType.category.

  There is no clear description when this value would be applicable. In fact, doubts are warranted whether the value (and with it, the attribute maxBaseTypeSize) should be applicable in AUTOSAR at all.

Background: the definition of a SwBaseType is supposed to refer to an atomic data type. The "base type" for an array is still the "base type" of the array elements, as opposed to the definition of a base type for the array itself.

The difference between the definition of a "base type" for the array elements and a "base type" for the entire array manifests in the attributes baseTypeSize, resp. maxBaseTypeSize.

In some cases definitions of SwBaseType can be found in the wild that look similar to this specimen:

```
<SW-BASE-TYPE>
<SHORT-NAME>FunnyBaseType</SHORT-NAME>
<CATEGORY>VARIABLE_LENGTH</CATEGORY>
<MAX-BASE-TYPE-SIZE>1433</MAX-BASE-TYPE-SIZE>
<BASE-TYPE-ENCODING>WINDOWS-1252</BASE-TYPE-ENCODING>
</SW-BASE-TYPE>
```

I guess that this base type has been build on the assumption that a string deserves a base type for the entire Windows-1252-encoded string.

This assumption would not only result in countless definitions of SwBaseTypes for each possible combination of array length and data type, it would also create an unholy dependency to the value of ImplementationDataTypeElement.arraySize. In other words, if the value of array size changes than the value of maxBaseTypeSize would have to change accordingly.

This does not make much sense.

I tried to get some information about the use case for VARIABLE_LENGTH that was originally intended. Bernhard mentioned that it was meant to support edge cases for the generation of A2L files, in particular for string implementations where the first element denotes the length and the rest is reserved for the payload.

If there are further known use cases, please feel free to leave a comment ...

I would like to provide a better explanation to the applicability of VARIABLE_LENGTH and also better explain that the SwBaseType is not meant to be applied to composite data types.

At the moment the document is rather indifferent about these two aspects.

Agreed solution:

SWCT:
—

1. Reformulate:

[constr_1011] category of SwBaseType
d The attribute SwBaseType.category must be set and only the values FIXED_LENGTH and VOID are supported. c

[constr_1422] Value of category is VOID
d If the value of the attribute SwBaseType.category is set to VOID then the attribute baseTypeSize shall not exist. c()

[constr_1012] Value of category is FIXED_LENGTH
d If the value of the attribute SwBaseType.category is set to FIXED_LENGTH then the attribute baseTypeSize shall be filled with content. c()

[constr_1229] category of ImplementationDataType boils down to VALUE
subElement.category is set to VALUE or TYPE_REFERENCE that eventually boils down to VALUE and the subElement refers to a SwBaseType where baseTypeSize is set to the value 8 and the baseTypeEncoding is set to NONE.

[constr_1220] Compatibility of SwBaseType
d Two SwBaseTypes are compatible if and only if attributes baseTypeSize respectively byteOrder, memAlignment, baseTypeEncoding, and nativeDeclaration have identical values. c()

2. Remove [constr_1013].

3. Set attribute maxBaseTypeSize to ’obsolete’.

4. Adapt Listing 5.4: Example for the definition of a string SwBaseType: (add CATEGORY)

```
<AR-PACKAGE>
<ELEMENTS>
<SW-BASE-TYPE>
<SHORT-NAME>MyTextBaseType</SHORT-NAME>
<CATEGORY>FIXED_LENGTH</CATEGORY>
<BASE-TYPE-SIZE>8</BASE-TYPE-SIZE>
<BASE-TYPE-ENCODING>UTF-8</BASE-TYPE-ENCODING>
```

</SW-BASE-TYPE>
<SW-BASE-TYPE>
<SHORT-NAME>uint8BaseType</SHORT-NAME>
<CATEGORY>FIXED_LENGTH</CATEGORY>
<BASE-TYPE-SIZE>8</BASE-TYPE-SIZE>
</SW-BASE-TYPE>
</ELEMENTS>
</AR-PACKAGE>

5. Fix all object diagrams showing SwBaseType (add CATEGORY).

DEXT:
—–

1. Remove in Table 4.8: Allowed attributes of SwDataDefProps for DiagnosticDataElement.swDataDefProps the line baseType.baseTypeDefinition.maxBaseTypeSize N/A.
–Last change on issue 74446 comment 7–

BW-C-Level:

Application	Specification	Bus
3	3	1

1.14 Specification Item constr_1229

Trace References:

none

Content:

An ImplementationDataType qualifies as an Integral Primitive Type if and only if either

- its Identifiable.category is VALUE or TYPE_REFERENCE that eventually boils down to VALUEor
- its Identifiable.category is ARRAYand it has only one ImplementationDataType.sub Elementand one of the following conditions applies:
  - ImplementationDataType.subElement.Identifiable.category is set to VALUE or TYPE_REFERENCE that eventually boils down to VALUEand the ImplementationDataType.subElement refers to a SwBaseType where BaseTypeDirect Definition.baseTypeSize or BaseTypeDirectDefinition.maxBaseTypeSize is set

to the value 8 and the BaseTypeDirectDefinition.baseTypeEncoding is set to NONE.

  - ImplementationDataType.subElement.Identifiable.category is set to TYPE_REFERENCEand the ImplementationDataTypeElement.swDataDefProps.SwDataDefProps.implementationDataType literally represents the Platform Data Type named “uint8”.
  - ImplementationDataType.subElement.Identifiable.category is set to TYPE_REFERENCEand the attribute ImplementationDataTypeElement.swDataDefProps.SwDataDefProps.implementationDataType.Referrable.shortName is set to “uint8” andAutosarDataType.swDataDefProps.SwDataDefProps.baseType.BaseType.baseTypeDefinition.BaseTypeDirectDefinition.nativeDeclaration does not exist.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #74446: Explanation for the applicability of VARIABLE_LENGTH missing

  Problem description:

  The SWCT contains constr_1011 and constr_1013 that mention the value VARIABLE_LENGTH as a possible setting for SwBaseType.category.

  There is no clear description when this value would be applicable. In fact, doubts are warranted whether the value (and with it, the attribute maxBaseTypeSize) should be applicable in AUTOSAR at all.

  Background: the definition of a SwBaseType is supposed to refer to an atomic data type. The "base type" for an array is still the "base type" of the array elements, as opposed to the definition of a base type for the array itself.

  The difference between the definition of a "base type" for the array elements and a "base type" for the entire array manifests in the attributes baseTypeSize, resp. maxBaseTypeSize.

  In some cases definitions of SwBaseType can be found in the wild that look similar to this specimen:

```
<SW-BASE-TYPE>
<SHORT-NAME>FunnyBaseType</SHORT-NAME>
<CATEGORY>VARIABLE_LENGTH</CATEGORY>
<MAX-BASE-TYPE-SIZE>1433</MAX-BASE-TYPE-SIZE>
<BASE-TYPE-ENCODING>WINDOWS-1252</BASE-TYPE-ENCODING>
</SW-BASE-TYPE>
```

I guess that this base type has been build on the assumption that a string deserves a base type for the entire Windows-1252-encoded string.

This assumption would not only result in countless definitions of SwBaseTypes for each possible combination of array length and data type, it would also create an unholy dependency to the value of ImplementationDataTypeElement.arraySize. In other words, if the value of array size changes than the value of maxBaseTypeSize would have to change accordingly.

This does not make much sense.

I tried to get some information about the use case for VARIABLE_LENGTH that was originally intended. Bernhard mentioned that it was meant to support edge cases for the generation of A2L files, in particular for string implementations where the first element denotes the length and the rest is reserved for the payload.

If there are further known use cases, please feel free to leave a comment ...

I would like to provide a better explanation to the applicability of VARIABLE_LENGTH and also better explain that the SwBaseType is not meant to be applied to composite data types.

At the moment the document is rather indifferent about these two aspects.

Agreed solution:

SWCT:
—–

1. Reformulate:

[constr_1011] category of SwBaseType
d The attribute SwBaseType.category must be set and only the values FIXED_LENGTH and VOID are supported. c

[constr_1422] Value of category is VOID
d If the value of the attribute SwBaseType.category is set to VOID then the attribute baseTypeSize shall not exist. c()

[constr_1012] Value of category is FIXED_LENGTH
d If the value of the attribute SwBaseType.category is set to FIXED_LENGTH then the attribute baseTypeSize shall be filled with content. c()

[constr_1229] category of ImplementationDataType boils down to VALUE
subElement.category is set to VALUE or TYPE_REFERENCE that eventually boils down to VALUE and the subElement refers to a SwBaseType where baseTypeSize is set to the value 8 and the baseTypeEncoding is set to NONE.

[constr_1220] Compatibility of SwBaseType
d Two SwBaseTypes are compatible if and only if attributes baseTypeSize respectively byteOrder, memAlignment, baseTypeEncoding, and nativeDeclaration have identical values. c()

2. Remove [constr_1013].

3. Set attribute maxBaseTypeSize to ’obsolete’.

4. Adapt Listing 5.4: Example for the definition of a string SwBaseType: (add CATEGORY)

```
<AR-PACKAGE>
<ELEMENTS>
<SW-BASE-TYPE>
<SHORT-NAME>MyTextBaseType</SHORT-NAME>
<CATEGORY>FIXED_LENGTH</CATEGORY>
<BASE-TYPE-SIZE>8</BASE-TYPE-SIZE>
<BASE-TYPE-ENCODING>UTF-8</BASE-TYPE-ENCODING>
</SW-BASE-TYPE>
<SW-BASE-TYPE>
<SHORT-NAME>uint8BaseType</SHORT-NAME>
<CATEGORY>FIXED_LENGTH</CATEGORY>
<BASE-TYPE-SIZE>8</BASE-TYPE-SIZE>
</SW-BASE-TYPE>
</ELEMENTS>
</AR-PACKAGE>
```

5. Fix all object diagrams showing SwBaseType (add CATEGORY).

DEXT:
—–

1. Remove in Table 4.8: Allowed attributes of SwDataDefProps for DiagnosticDataElement.swDataDefProps the line baseType.baseTypeDefinition.maxBaseTypeSize N/A.
–Last change on issue 74446 comment 7–

BW-C-Level:

Application	Specification	Bus
3	3	1

1.15 Specification Item constr_1288

Trace References:

none

Content:

The allowed values per Identifiable.category for DataPrototypes typed by ImplementationDataTypes are documented in table [REF table_3a_CategoriesImpl4DataProt]. 71082,76585

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #74212: Clarifications on DataConstrs w.r.t. optionallity and PhysConstrs versus InternalConstr

  Problem description:

  The figure Figure 5.6: Value ranges and invalid values shows very nice how the different data constraints needs to match to each other.

  1.) But the figure shows something like intern constrains of an ApplicationDataType. How des this match to TPS_SWCT_01288 which implies that for ApplicationDataTypes only PhysConstrs are applicable. Or is this a misinterpretation?

  2.) Since DataConstr are optional, I miss also some precise statement how the limits of a data type is defined in case of missing DataConstr.

  I would assume the following:
  If an ApplicationDataTypes does not define a DataConstr the limit of the later mapped ImplementationDataType applies.
  If an ImplementationDataType does not define a DataConstr the technical limit of the BaseType defined by the size of type in bits and the according encoding applies.

  Agreed solution:

1. Split the dataConstr line in tables 5.8, 5.18, 5.32, and 5.33 such that two new line are created

dataConstr.physConstr
dataConstr.internalConstr

Keep the existing setting for the respective flavor (application vs. implementation) and set the respective other line to to "d/c" and add an explanation to the legend that "d/c" stands for "don't care.

Update TPS_SWCT_01288:

[TPS_SWCT_01288] Interpretation of PhysConstrs and InternalConstrs by tools
DataConstr is an ARElement which can be reused by several data type specifications. Especially an ImplementationDataType and an Application-
DataType which are mapped to each other, can refer to the same constraints or they can define their own constraints.
To avoid conflicts, in both cases PhysConstrs shall be interpreted by tools only with respect to application data types while InternalConstrs shall be interpreted only with respect to implementation data types.
In case only physConstr are provided to ApplicationDataTypes the CompuMethod can be used to compute the internalConstr. ()

In Figure 5.6 (Value ranges and invalid values), replace the string "internalConstrs of ApplicationDataType" by "computed internalConstrs of ApplicationDataType"

2. Add note: If an ApplicationPrimitiveDataType does not define dataConstr then implicit constraints can be derived from physical meaning of the ApplicationDataType. For example, if the data type represents a temperature, the lower bound will not be able to exceed 0K. For other physical meanings it could be possible that the implicitly assumed limits go from -INF to +INF. In order to void ambiguity for the values of limits it is strongly recommended to define a reasonable limit for ApplicationPrimitiveDataTypes.
–Last change on issue 74212 comment 7–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.16 Specification Item constr_1289

Trace References:

none

Content:

The allowed values of Attributes per Identifiable.category for DataPrototypes typed by ApplicationDataTypes are documented in table [REF table_3a_CategoriesAppl4DataProt].72038,72877,76585

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #74212: Clarifications on DataConstrs w.r.t. optionallity and PhysConstrs versus InternalConstr

  Problem description:

  The figure Figure 5.6: Value ranges and invalid values shows very nice how the different data constraints needs to match to each other.

  1.) But the figure shows something like intern constrains of an ApplicationDataType. How des this match to TPS_SWCT_01288 which implies that for ApplicationDataTypes only PhysConstrs are applicable. Or is this a misinterpretation?

  2.) Since DataConstr are optional, I miss also some precise statement how the limits of a data type is defined in case of missing DataConstr.

  I would assume the following:
  If an ApplicationDataTypes does not define a DataConstr the limit of the later mapped ImplementationDataType applies.
  If an ImplementationDataType does not define a DataConstr the technical limit of the BaseType defined by the size of type in bits and the according encoding applies.

  Agreed solution:

  1. Split the dataConstr line in tables 5.8, 5.18, 5.32, and 5.33 such that two new line are created

  dataConstr.physConstr
  dataConstr.internalConstr

  Keep the existing setting for the respective flavor (application vs. implementation) and set the respective other line to to "d/c" and add an explanation to the legend that "d/c" stands for "don't care.

Update TPS_SWCT_01288:

[TPS_SWCT_01288] Interpretation of PhysConstrs and InternalConstrs by tools
DataConstr is an ARElement which can be reused by several data type specifications. Especially an ImplementationDataType and an Application-
DataType which are mapped to each other, can refer to the same constraints or they can define their own constraints.
To avoid conflicts, in both cases PhysConstrs shall be interpreted by tools only with respect to application data types while InternalConstrs shall be interpreted only with respect to implementation data types.
In case only physConstr are provided to ApplicationDataTypes the CompuMethod can be used to compute the internalConstr. ()

In Figure 5.6 (Value ranges and invalid values), replace the string "internalConstrs of ApplicationDataType" by "computed internalConstrs of ApplicationDataType"

2. Add note: If an ApplicationPrimitiveDataType does not define dataConstr then implicit constraints can be derived from physical meaning of the ApplicationDataType. For example, if the data type represents a temperature, the lower bound will not be able to exceed 0K. For other physical meanings it could be possible that the implicitly assumed limits go from -INF to +INF. In order to void ambiguity for the values of limits it is strongly recommended to define a reasonable limit for ApplicationPrimitiveDataTypes.
–Last change on issue 74212 comment 7–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.17 Specification Item constr_1422

Trace References:

none

Content:

If the value of the attribute SwBaseType.Identifiable.category is set to VOID then neither the attribute BaseTypeDirectDefinition.baseTypeSize nor the attribute BaseTypeDirectDefinition.maxBaseTypeSize shall shall not exist.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #74446: Explanation for the applicability of VARIABLE_LENGTH missing

  Problem description:

  The SWCT contains constr_1011 and constr_1013 that mention the value VARIABLE_LENGTH as a possible setting for SwBaseType.category.

  There is no clear description when this value would be applicable. In fact, doubts are warranted whether the value (and with it, the attribute maxBaseTypeSize) should be applicable in AUTOSAR at all.

  Background: the definition of a SwBaseType is supposed to refer to an atomic data type. The "base type" for an array is still the "base type" of the array elements, as opposed to the definition of a base type for the array itself.

  The difference between the definition of a "base type" for the array elements and a "base type" for the entire array manifests in the attributes baseTypeSize, resp. maxBaseTypeSize.

  In some cases definitions of SwBaseType can be found in the wild that look similar to this specimen:

  ```
  <SW-BASE-TYPE>
  <SHORT-NAME>FunnyBaseType</SHORT-NAME>
  <CATEGORY>VARIABLE_LENGTH</CATEGORY>
  <MAX-BASE-TYPE-SIZE>1433</MAX-BASE-TYPE-SIZE>
  <BASE-TYPE-ENCODING>WINDOWS-1252</BASE-TYPE-ENCODING>
  </SW-BASE-TYPE>
  ```

  I guess that this base type has been build on the assumption that a string deserves a base type for the entire Windows-1252-encoded string.

  This assumption would not only result in countless definitions of SwBaseTypes for each possible combination of array length and data type, it would also create an unholy dependency to the value of ImplementationDataTypeElement.arraySize. In other words, if the value of array size changes than the value of maxBaseTypeSize would have to change accordingly.

  This does not make much sense.

  I tried to get some information about the use case for VARIABLE_LENGTH that was originally intended. Bernhard mentioned that it was meant to support edge cases for the generation of A2L files, in particular for string implementations where the first element denotes the length and the rest is reserved for the payload.

If there are further known use cases, please feel free to leave a comment ...

I would like to provide a better explanation to the applicability of VARIABLE_LENGTH and also better explain that the SwBaseType is not meant to be applied to composite data types.

At the moment the document is rather indifferent about these two aspects.

Agreed solution:

SWCT:
—–

1. Reformulate:

[constr_1011] category of SwBaseType
d The attribute SwBaseType.category must be set and only the values FIXED_LENGTH and VOID are supported. c

[constr_1422] Value of category is VOID
d If the value of the attribute SwBaseType.category is set to VOID then the attribute baseTypeSize shall not exist. c()

[constr_1012] Value of category is FIXED_LENGTH
d If the value of the attribute SwBaseType.category is set to FIXED_LENGTH then the attribute baseTypeSize shall be filled with content. c()

[constr_1229] category of ImplementationDataType boils down to VALUE
subElement.category is set to VALUE or TYPE_REFERENCE that eventually boils down to VALUE and the subElement refers to a SwBaseType where baseTypeSize is set to the value 8 and the baseTypeEncoding is set to NONE.

[constr_1220] Compatibility of SwBaseType
d Two SwBaseTypes are compatible if and only if attributes baseTypeSize respectively byteOrder, memAlignment, baseTypeEncoding, and nativeDeclaration have identical values. c()

2. Remove [constr_1013].

3. Set attribute maxBaseTypeSize to ’obsolete’.

4. Adapt Listing 5.4: Example for the definition of a string SwBaseType: (add CATEGORY)

```
<AR-PACKAGE>
<ELEMENTS>
<SW-BASE-TYPE>
<SHORT-NAME>MyTextBaseType</SHORT-NAME>
<CATEGORY>FIXED_LENGTH</CATEGORY>
<BASE-TYPE-SIZE>8</BASE-TYPE-SIZE>
<BASE-TYPE-ENCODING>UTF-8</BASE-TYPE-ENCODING>
</SW-BASE-TYPE>
<SW-BASE-TYPE>
<SHORT-NAME>uint8BaseType</SHORT-NAME>
<CATEGORY>FIXED_LENGTH</CATEGORY>
<BASE-TYPE-SIZE>8</BASE-TYPE-SIZE>
</SW-BASE-TYPE>
</ELEMENTS>
</AR-PACKAGE>
```

5. Fix all object diagrams showing SwBaseType (add CATEGORY).

DEXT:
—

1. Remove in Table 4.8: Allowed attributes of SwDataDefProps for DiagnosticDataElement.swDataDefProps the line baseType.baseTypeDefinition.maxBaseTypeSize N/A.
–Last change on issue 74446 comment 7–

BW-C-Level:

Application	Specification	Bus
3	3	1

1.18 Specification Item constr_1436

Trace References:

none

Content:

If the attribute DiagnosticCommunicationManagerNeeds.DiagnosticCommunicationManagerNeeds.serviceRequestCallbackType exist and is set to DiagnosticServiceRequest CallbackTypeEnum.requestCallbackTypeSupplier then parameter DcmDsdRequestSupplierNotificationEnabled shall be set to TRUE and container DcmDsdServiceRequest SupplierNotification shall exist and the value of DcmDsdServiceRequestSupplierNotification.Referrable.shortName shall be taken from the Referrable.shortName of the enclosing SwcServiceDependency.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #74513: How to handle EcuC constraints in template documents

  Problem description:

  The software component template contains two constraints which are referencing deep internals of the Dcm configuration.
  On one hand those constraints can be seen as the extended version of upstream mappings. On the other hand its a bit strange that the SWC-T contains such deep internals of the Dcm Ecuc.

  I'm a bit concerned that this information is not expected at this place.

  [constr_1436] DiagnosticCommunicationManagerNeeds.serviceRequest-
  CallbackType is set to requestCallbackTypeSupplier d If the attribute
  DiagnosticCommunicationManagerNeeds.serviceRequestCallbackType
  exist and is set to requestCallbackTypeSupplier then parameter DcmDsdRequest-
  SupplierNotificationEnabled
  shall be set to TRUE and container
  DcmDsdServiceRequestSupplierNotification shall exist and the value of
  DcmDsdServiceRequestSupplierNotification.shortName shall be taken
  from the shortName of the enclosing SwcServiceDependency. c()

  [constr_1437] DiagnosticCommunicationManagerNeeds.serviceRequest-
  CallbackType is set to requestCallbackTypeManufacturer d if the attribute
  DiagnosticCommunicationManagerNeeds.serviceRequestCallbackType
  exist and is set to requestCallbackTypeManufacturer then parameter DcmDs-
  dRequestManufacturerNotificationEnabled
  shall be set to TRUE and
  container DcmDsdServiceRequestManufacturerNotification shall exist and
  the value of DcmDsdServiceRequestManufacturerNotification.shortName
  shall be taken from the shortName of the enclosing SwcServiceDependency. c()

Agreed solution:

Upstream Mappings:

DcmDsdRequestSupplierNotificationEnabled –> DiagnosticCommunicationManagerNeeds.serviceRequestCallbackType (full, "If DiagnosticCommunicationManagerNeeds.serviceRequestCallbackType is set to DiagnosticServiceRequestCallbackTypeEnum.requestCallbackTypeSupplier then DcmDsdRequestSupplierNotificationEnabled shall be set to TRUE", TPS_SWCT)

DcmDsdServiceRequestSupplierNotification –> DiagnosticCommunicationManagerNeeds.serviceRequestCallbackType (full, "If DiagnosticCommunicationManagerNeeds.serviceRequestCallbackType is set to DiagnosticServiceRequestCallbackTypeEnum.requestCallbackTypeSupplier then DcmDsdServiceRequestSupplierNotification shall exist and the value of DcmDsdServiceRequestSupplierNotification.shortName shall be taken from the SwcServiceDependency.shortName that aggregates the DiagnosticCommunicationManagerNeeds", TPS_SWCT)

DcmDsdRequestManufacturerNotificationEnabled –> DiagnosticCommunicationManagerNeeds.serviceRequestCallbackType (full, "If DiagnosticCommunicationManagerNeeds.serviceRequestCallbackType is set to DiagnosticServiceRequestCallbackTypeEnum.requestCallbackTypeManufacturer then DcmDsdRequestManufacturerNotificationEnabled shall be set to TRUE", TPS_SWCT)

DcmDsdServiceRequestManufacturerNotification –> DiagnosticCommunicationManagerNeeds.serviceRequestCallbackType (full, "If DiagnosticCommunicationManagerNeeds.serviceRequestCallbackType is set to DiagnosticServiceRequestCallbackTypeEnum.requestCallbackTypeManufacturer then DcmDsdServiceRequestManufacturerNotification shall exist and the value of DcmDsdServiceRequestManufacturerNotification.shortName shall be taken from the SwcServiceDependency.shortName that aggregates the DiagnosticCommunicationManagerNeeds", TPS_SWCT)

remove constr_1436 and constr_1437
–Last change on issue 74513 comment 8–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.19 Specification Item constr_1437

Trace References:

none

Content:

if the attribute DiagnosticCommunicationManagerNeeds.DiagnosticCommunicationManagerNeeds.serviceRequestCallbackType exist and is set to DiagnosticServiceRequest CallbackTypeEnum.requestCallbackTypeManufacturer then parameter DcmDsdRequest ManufacturerNotificationEnabled shall be set to TRUE and container DcmDsdServiceRequestManufacturerNotification shall exist and the value of DcmDsdServiceRequestManufacturerNotification.Referrable.shortName shall be taken from the Referrable.shortName of the enclosing SwcServiceDependency.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #74513: How to handle EcuC constraints in template documents

  Problem description:

  The software component template contains two constraints which are referencing deep internals of the Dcm configuration.
  On one hand those constraints can be seen as the extended version of upstream mappings. On the other hand its a bit strange that the SWC-T contains such deep internals of the Dcm Ecuc.

  I'm a bit concerned that this information is not expected at this place.

  [constr_1436] DiagnosticCommunicationManagerNeeds.serviceRequest-CallbackType is set to requestCallbackTypeSupplier d If the attribute
  DiagnosticCommunicationManagerNeeds.serviceRequestCallbackType
  exist and is set to requestCallbackTypeSupplier then parameter DcmDsdRequest-SupplierNotificationEnabled
  shall be set to TRUE and container
  DcmDsdServiceRequestSupplierNotification shall exist and the value of
  DcmDsdServiceRequestSupplierNotification.shortName shall be taken
  from the shortName of the enclosing SwcServiceDependency. c()

  [constr_1437] DiagnosticCommunicationManagerNeeds.serviceRequest-CallbackType is set to requestCallbackTypeManufacturer d if the attribute
  DiagnosticCommunicationManagerNeeds.serviceRequestCallbackType

exist and is set to requestCallbackTypeManufacturer then parameter DcmDsdRequestManufacturerNotificationEnabled
shall be set to TRUE and
container DcmDsdServiceRequestManufacturerNotification shall exist and
the value of DcmDsdServiceRequestManufacturerNotification.shortName
shall be taken from the shortName of the enclosing SwcServiceDependency. c()

Agreed solution:

Upstream Mappings:

DcmDsdRequestSupplierNotificationEnabled –> DiagnosticCommunicationManagerNeeds.serviceRequestCallbackType (full, "If DiagnosticCommunicationManagerNeeds.serviceRequestCallbackType is set to DiagnosticServiceRequestCallbackTypeEnum.requestCallbackTypeSupplier then DcmDsdRequestSupplierNotificationEnabled shall be set to TRUE", TPS_SWCT)

DcmDsdServiceRequestSupplierNotification –> DiagnosticCommunicationManagerNeeds.serviceRequestCallbackType (full, "If DiagnosticCommunicationManagerNeeds.serviceRequestCallbackType is set to DiagnosticServiceRequestCallbackTypeEnum.requestCallbackTypeSupplier then DcmDsdServiceRequestSupplierNotification shall exist and the value of DcmDsdServiceRequestSupplierNotification.shortName shall be taken from the SwcServiceDependency.shortName that aggregates the DiagnosticCommunicationManagerNeeds", TPS_SWCT)

DcmDsdRequestManufacturerNotificationEnabled –> DiagnosticCommunicationManagerNeeds.serviceRequestCallbackType (full, "If DiagnosticCommunicationManagerNeeds.serviceRequestCallbackType is set to DiagnosticServiceRequestCallbackTypeEnum.requestCallbackTypeManufacturer then DcmDsdRequestManufacturerNotificationEnabled shall be set to TRUE", TPS_SWCT)

DcmDsdServiceRequestManufacturerNotification –> DiagnosticCommunicationManagerNeeds.serviceRequestCallbackType (full, "If DiagnosticCommunicationManagerNeeds.serviceRequestCallbackType is set to DiagnosticServiceRequestCallbackTypeEnum.requestCallbackTypeManufacturer then DcmDsdServiceRequestManufacturerNotification shall exist and the value of DcmDsdServiceRequestManufacturerNotification.shortName shall be taken from the SwcServiceDependency.shortName that aggregates the DiagnosticCommunicationManagerNeeds", TPS_SWCT)

remove constr_1436 and constr_1437
–Last change on issue 74513 comment 8–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.20 Specification Item constr_1515

Trace References:

none

Content:

For any given ImplementationDataTypeSubElementRef, either the aggregation

- ImplementationDataTypeSubElementRef.ImplementationDataTypeSubElementRef.implementationDataTypeElement or
- ImplementationDataTypeSubElementRef.ImplementationDataTypeSubElementRef.parameterImplementationDataTypeElement

shall exist.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76467: SubElementMapping for ParameterDataProtypes typed by ImplementationDataType

  Problem description:

  VariableAndParameterInterfaceMapping breeds DataPrototypeMapping.
  DataPrototypeMapping breeds SubElementMapping.
  SubElementMapping breeds ImplementationDataTypeSubElementRef.

  And ImplementationDataTypeSubElementRef is the path to the dark side.

  Because ...

  ImplementationDataTypeSubElementRef refers to VariableDataPrototype in the role rootVariableDataPrototype.

  ImplementationDataTypeSubElementRef does NOT refer to ParameterDataPrototype in an applicable role to be used as the root element.

  This rules out the application of the ImplementationDataTypeSubElementRef for a ParameterDataPrototype that takes the role of a root element.

And again, what we actually wanted to implement with this mess is called VariableAnd**Parameter**InterfaceMapping.

Sigh.

Agreed solution:

Add new meta-class named ArParameterInImplementationDataInstanceRef (decription: <come up something creative out of the description of the ArVariableInImplementationDataInstanceRef>)

Aggregate ArParameterInImplementationDataInstanceRef at ImplementationDataTypeSubElementRef with the mulitplicity 0..1 in the role parameterImplementationDataTypeElement.

Change the multiplicity of the aggregation ImplementationDataTypeSubElementRef.implementationDataTypeElement to 0..1

Add reference from ArParameterInImplementationDataInstanceRef to ParameterDatapPototype in the role rootParameterDataPrototype. Multiplicity: 0..1

Add reference from ArParameterInImplementationDataInstanceRef to ImplementationDataTypeElement in the role contextDataPrototype. Multiplicity: 0..*

Add reference from ArParameterInImplementationDataInstanceRef to ImplementationDataTypeElement in the role targetDataPrototype. Mulitplicity: 1

add constraint:

constr_xxxx Existence of ImplementationDataTypeSubElementRef.implementationDataTypeElement vs. ImplementationDataTypeSubElementRef.parameterImplementationDataTypeElement
For any given ImplementationDataTypeSubElementRef, either the aggregation ImplementationDataTypeSubElementRef.implementationDataTypeElement or ImplementationDataTypeSubElementRef.parameterImplementationDataTypeElement shall exist.
–Last change on issue 76467 comment 5–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.21 Specification Item constr_1516

Trace References:

none

Content:

The reference ArParameterInImplementationDataInstanceRef.ArParameterInImplementationDataInstanceRef.contextDataPrototype shall be defined for

- each leaf (i.e. the end of a chain of aggregating elements) ImplementationDataType Element of Identifiable.categoryTYPE_REFERENCE in a chain of referencing ImplementationDataTypes which is not the ArParameterInImplementationDataInstance Ref.targetDataPrototype
- and each ImplementationDataTypeElement owned by an ImplementationDataType or ImplementationDataTypeElement of Identifiable.categoryARRAY in a chain of referencing ImplementationDataTypes

starting from the ImplementationDataTypes of the ArParameterInImplementationData InstanceRef.rootParameterDataPrototype down to the leaf ImplementationDataTypeElement which is typed (directly or indirectly via ImplementationDataType of Identifiable.categoryTYPE_REFERENCE) by the ImplementationDataType of the ArParameter InImplementationDataInstanceRef.targetDataPrototype.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76467: SubElementMapping for ParameterDataProtypes typed by ImplementationDataType

  Problem description:

  VariableAndParameterInterfaceMapping breeds DataPrototypeMapping.
  DataPrototypeMapping breeds SubElementMapping.
  SubElementMapping breeds ImplementationDataTypeSubElementRef.

  And ImplementationDataTypeSubElementRef is the path to the dark side.

  Because ...

  ImplementationDataTypeSubElementRef refers to VariableDataPrototype in the role rootVariableDataPrototype.

  ImplementationDataTypeSubElementRef does NOT refer to ParameterDataPrototype in an applicable role to be used as the root element.

This rules out the application of the ImplementationDataTypeSubElementRef for a ParameterDataPrototype that takes the role of a root element.

And again, what we actually wanted to implement with this mess is called VariableAnd**Parameter**InterfaceMapping.

Sigh.

Agreed solution:

Add new meta-class named ArParameterInImplementationDataInstanceRef (decription: <come up something creative out of the description of the ArVariableInImplementationDataInstanceRef>)

Aggregate ArParameterInImplementationDataInstanceRef at ImplementationDataTypeSubElementRef with the mulitplicity 0..1 in the role parameterImplementationDataTypeElement.

Change the multiplicity of the aggregation ImplementationDataTypeSubElementRef.implementationDataTypeElement to 0..1

Add reference from ArParameterInImplementationDataInstanceRef to ParameterDatapPototype in the role rootParameterDataPrototype. Multiplicity: 0..1

Add reference from ArParameterInImplementationDataInstanceRef to ImplementationDataTypeElement in the role contextDataPrototype. Multiplicity: 0..*

Add reference from ArParameterInImplementationDataInstanceRef to ImplementationDataTypeElement in the role targetDataPrototype. Mulitplicity: 1

add constraint:

constr_xxxx Existence of ImplementationDataTypeSubElementRef.implementationDataTypeElement vs. ImplementationDataTypeSubElementRef.parameterImplementationDataTypeElement
For any given ImplementationDataTypeSubElementRef, either the aggregation ImplementationDataTypeSubElementRef.implementationDataTypeElement or ImplementationDataTypeSubElementRef.parameterImplementationDataTypeElement shall exist.
–Last change on issue 76467 comment 5–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.22 Specification Item constr_1517

Trace References:

none

Content:

The attribute ArParameterInImplementationDataInstanceRef.ArParameterInImplementationDataInstanceRef.contextDataPrototype shall only exist for an ImplementationData TypeElementIdentifiable.categoryTYPE_REFERENCE or ARRAY.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76467: SubElementMapping for ParameterDataProtypes typed by ImplementationDataType

  Problem description:

  VariableAndParameterInterfaceMapping breeds DataPrototypeMapping.
  DataPrototypeMapping breeds SubElementMapping.
  SubElementMapping breeds ImplementationDataTypeSubElementRef.

  And ImplementationDataTypeSubElementRef is the path to the dark side.

  Because ...

  ImplementationDataTypeSubElementRef refers to VariableDataPrototype in the role rootVariableDataPrototype.

  ImplementationDataTypeSubElementRef does NOT refer to ParameterDataPrototype in an applicable role to be used as the root element.

  This rules out the application of the ImplementationDataTypeSubElementRef for a ParameterDataPrototype that takes the role of a root element.

  And again, what we actually wanted to implement with this mess is called VariableAnd**Parameter**InterfaceMapping.

  Sigh.

  Agreed solution:

  Add new meta-class named ArParameterInImplementationDataInstanceRef (decription: <come up something creative out of the description of the ArVariableInImplementationDataInstanceRef>)

Aggregate ArParameterInImplementationDataInstanceRef at ImplementationDataTypeSubElementRef with the mulitplicity 0..1 in the role parameterImplementationDataTypeElement.

Change the multiplicity of the aggregation ImplementationDataTypeSubElementRef.implementationDataTypeElement to 0..1

Add reference from ArParameterInImplementationDataInstanceRef to ParameterDatapPototype in the role rootParameterDataPrototype. Multiplicity: 0..1

Add reference from ArParameterInImplementationDataInstanceRef to ImplementationDataTypeElement in the role contextDataPrototype. Multiplicity: 0..*

Add reference from ArParameterInImplementationDataInstanceRef to ImplementationDataTypeElement in the role targetDataPrototype. Mulitplicity: 1

add constraint:

constr_xxxx Existence of ImplementationDataTypeSubElementRef.implementationDataTypeElement vs. ImplementationDataTypeSubElementRef.parameterImplementationDataTypeElement
For any given ImplementationDataTypeSubElementRef, either the aggregation ImplementationDataTypeSubElementRef.implementationDataTypeElement or ImplementationDataTypeSubElementRef.parameterImplementationDataTypeElement shall exist.
–Last change on issue 76467 comment 5–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.23 Specification Item constr_1518

Trace References:

none

Content:

The definition of attributes ArParameterInImplementationDataInstanceRef.contextData Prototype and ArParameterInImplementationDataInstanceRef.targetDataPrototype shall be enclosed in the context of the definition of the data type used to type ArParameterInImplementationDataInstanceRef.rootParameterDataPrototype.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76467: SubElementMapping for ParameterDataProtypes typed by ImplementationDataType

  Problem description:

  VariableAndParameterInterfaceMapping breeds DataPrototypeMapping.
  DataPrototypeMapping breeds SubElementMapping.
  SubElementMapping breeds ImplementationDataTypeSubElementRef.

  And ImplementationDataTypeSubElementRef is the path to the dark side.

  Because ...

  ImplementationDataTypeSubElementRef refers to VariableDataPrototype in the role rootVariableDataPrototype.

  ImplementationDataTypeSubElementRef does NOT refer to ParameterDataPrototype in an applicable role to be used as the root element.

  This rules out the application of the ImplementationDataTypeSubElementRef for a ParameterDataPrototype that takes the role of a root element.

  And again, what we actually wanted to implement with this mess is called VariableAnd**Parameter**InterfaceMapping.

  Sigh.

  Agreed solution:

  Add new meta-class named ArParameterInImplementationDataInstanceRef (decription: <come up something creative out of the description of the ArVariableInImplementationDataInstanceRef>)

  Aggregate ArParameterInImplementationDataInstanceRef at ImplementationDataTypeSubElementRef with the mulitplicity 0..1 in the role parameterImplementationDataTypeElement.

  Change the multiplicity of the aggregation ImplementationDataTypeSubElementRef.implementationDataTypeElement to 0..1

Add reference from ArParameterInImplementationDataInstanceRef to ParameterDatapPototype in the role rootParameterDataPrototype. Multiplicity: 0..1

Add reference from ArParameterInImplementationDataInstanceRef to ImplementationDataTypeElement in the role contextDataPrototype. Multiplicity: 0..*

Add reference from ArParameterInImplementationDataInstanceRef to ImplementationDataTypeElement in the role targetDataPrototype. Mulitplicity: 1

add constraint:

constr_xxxx Existence of ImplementationDataTypeSubElementRef.implementationDataTypeElement vs. ImplementationDataTypeSubElementRef.parameterImplementationDataTypeElement
For any given ImplementationDataTypeSubElementRef, either the aggregation ImplementationDataTypeSubElementRef.implementationDataTypeElement or ImplementationDataTypeSubElementRef.parameterImplementationDataTypeElement shall exist.
–Last change on issue 76467 comment 5–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.24 Specification Item constr_1519

Trace References:

none

Content:

The existence of attributes of meta-class ApplicationValueSpecification vs. the value of Identifiable.category is regulated by Table [REF fig_3a_AUTOSAR_Meta_2d_Model_Model_DOC_ApplicationValueSpecification].

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77041: Categories for ApplicationValueSpecifcation

  Problem description:

  constr_1006 describes the valid categories for various model elements, among those: ApplicationValueSpecification.

But we don't have a clear definition of how the categories should be used and in which case which category would be applicable.

I also do not see why the category VAL_BLK is allowed but ARRAY isn't. There is just no explanation available at all.

Agreed solution:

The multiplicity of SwAxisCont.unit and SwValueCont.unit shall be changed to 0..1.

Add the existing example of a MAP with two STD_AXIS to the document. In the example, the vf shall be replaced by v.

Add the sudoku attached to the RFC as a constraint to the document.
–Last change on issue 77041 comment 4–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.25 Specification Item constr_1520

Trace References:

none

Content:

In the context of an SwcServiceDependency, each DiagnosticEventNeeds referenced in the role ObdRatioServiceNeeds.rateBasedMonitoredEvent shall only be referenced by at most a single ObdRatioServiceNeeds.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77072: Remove source multiplicities and source roles from the model

  Problem description:

  Remove all source multiplicities from the model.

  Agreed solution:

  Please implement in 80_Common and merge with the CP and AP branches.

  TPS_SystemTemplate:

- the source multiplicity of the CommunicationController->CommunicationConnector association shall be removed (*).
- the source multiplicity of the ISignal->SystemSignal association shall be removed (1..*).
- the source multiplicity of the ISignalGroup->SystemSignalGroup association shall be removed (1..*).
- the source multiplicity of the FrameTriggering->Frame association shall be removed (1..*)
- the source multiplicity of the ISignalTriggering->ISignal association shall be removed (0..*)
- the source multiplicity of the ISignalIPduGroup->ISignalIPduGroup association-shall be removed (0..1).
- the source multiplicity of the PduTriggering->Pdu association shall be removed (1..*).
- the source multiplicity of the EcuInstance->ISignalIPduGroup association shall be removed (1).
- the source multiplicity of the NmNode->NmEcu association shall be removed (1..*).
- the source multiplicity of the CanNmClusterCoupling->CanNmCluster association shall be removed (0..1).
- the source multiplicity of the FlexrayNmClusterCoupling->FlexrayNmCluster association shall be removed (0..1).
- the source multiplicity of the UdpNmClusterCoupling->UdpNmCluster association shall be removed (0..1).
- the source multiplicity of the EcuResourceEstimation->EcuInstance association shall be removed (*)
- the source multiplicity of the EcuResourceEstimation->SwcToEcuMapping association shall be removed (*)
- the source multiplicity of the SignalPathConstraint->DocumentationBlock association shall be removed (0..1)
==
ECU Configuration specification:
- the source multiplicity of the EcucModuleConfigurationValues -> EcucContainerValue aggregation shall be removed (0..1)
- the source multiplicity of the EcucContainerValue -> EcucContainerValue aggregation shall be removed (0..1)
==
Ecu Resource Template:
- the source multiplicity of the HwElement -> HwElement association shall be removed (0..1)

==
Generic Structure Template:

- the source multiplicity of the RefConditional -> VariationPoint aggregation shall be removed (0)
==
Basic Software Module Description:

- the source multiplicity of the BswAsynchronousServerCallResultPoint -> BswAsynchronousServerCallPoint association shall be removed (1)
- the source multiplicity of the BswAsynchronousServerCallReturnsEvent -> BswAsynchronousServerCallResultPoint association shall be removed (1)
- the source multiplicity of the BswInternalTriggerOccurredEvent -> BswInternalTriggeringPoint association shall be removed (1..*)
==
SWCT:

- Remove source multiplicity 0..* from dependency from SwcModeSwitchEvent to ModeDeclaration in the target-role "mode"

- Remove source multiplicity 0..* from dependency from ModeAccessPoint to ModeDeclarationGroupPrototype in the target-role "modeGroup"

- Remove source multiplicity 1 and source role "interface" from aggregation from ModeSwitchInterface to ModeDeclarationGroupPrototype in the target-role "modeGroup", no constraint required because there is no reasonable multiplicity for aggregations other than 1 or 0..1.

- Remove source multiplicity 0..* from association from ModePortAnnotation to ModeDeclarationGroupPrototype in the target-role "modeGroup"

- Remove source multiplicity 0..* from dependency from ModeSwitchPoint to ModeDeclarationGroupPrototype in the target-role "modeGroup"

- Remove source multiplicity 0..* from association from TriggerPortAnnotation to Trigger in the target-role "trigger"

- Remove source role "event" from association from ExternalTriggerOccurredEvent to Trigger in the target-role "trigger"

- Remove source multiplicity 0..1 from association from ObdRatioServiceNeeds to DiagnosticEventNeeds in the target-role "rateBasedMonitoredEvent", add constraint:

[constr_xxxx] Semantics of ObdRatioServiceNeeds.rateBasedMonitoredEvent
In the context of an SwcServiceDependency, each DiagnostivEventNeeds referenced in the role rateBasedMonitoredEvent shall only be referenced by at most a

single ObdRatioServiceNeeds.

- Remove source role "interface" from aggregation from NvDataInterface to VariableDataPrototype in the target-role "nvData"

- Remove source role "interface" and multiplicity 1 from aggregation from SenderReceiverInterface to VariableDataPrototype in the target-role "nvData"

- Remove source multiplicity 0..* from association from NvDataPortAnnotation to VariableDataPrototype in the target-role "variable"

- Remove source multiplicity 0..1 from association from SenderReceiverAnnotation to VariableDataPrototype in the target-role "dataElement", no constraint needed because annotations represent such a weak semantics

- Remove source role "event" from dependency from DataReceiveErrorEvent to VariableDataPrototype in the target-role "data"

- Remove source role "event" from dependency from DataReceivedEvent to VariableDataPrototype in the target-role "data"

- Remove source multiplicity 0..* from association from AssemblySwConnector to AbstractProvidedPortPrototype in the target-role "provider"

- Remove source multiplicity 0..* from association from AssemblySwConnector to AbstractRequiredPortPrototype in the target-role "requester"

- Remove source multiplicity * from association from SwComponentPrototype to SwComponentType in the target-role "type"

- Remove source role "component" from aggregation from SwComponentType to PortPrototype in the target-role "port"

- Remove source role "pPort" and multiplicity * from association from PPortPrototype to PortInterface in the target-role "providedInterface"

- Remove source multiplicity 0..* from dependency from DelegationSwConnector to PortPrototype in the target-role "innerPort"

- Remove source multiplicity 0..1 from association from PortAPIOption to PortPrototype in the target-role "port", no constraint needed because annotations represent such a weak semantics

- Remove source multiplicity 0..* from association from DelegationSwConnector to PortPrototype in the target-role "outerPort"

- Remove source role "rPort" and multiplicity * from association from RPortPrototype to PortInterface in the target-role "requiredInterface"

- Remove source multiplicity 0..1 from aggregation from ModeSwitchPoint to PModeGroupInAtomicSwcInstanceRef in the target-role "modeGroup"

- Remove source multiplicity 1 and source role "event" from aggregation from OperationInvokedEvent to POperationInAtomicSwcInstanceRef in the target-role "operation"

- Remove source multiplicity 1..* and source role "event" from aggregation from ExternalTriggerOccurredEvent to RTriggerInAtomicSwcInstanceRef in the target-role "trigger"

- Remove source multiplicity 0..1 and source role "event" from aggregation from DataReceiveErrorEvent to RVariableInAtomicSwcInstanceRef in the target-role "data"

- Remove source multiplicity * and source role "event" from aggregation from DataReceivedEvent to RVariableInAtomicSwcInstanceRef in the target-role "data"

- Remove source role "composition" from aggregation from CompositionSwComponentType to SwConnector in the target-role "connector"

- Remove source multiplicity 1 and source role "interface" from aggregation from ClientServerInterface to ClientServerOperation in the target-role "operation"

- Remove source multiplicity 1 from aggregation from ClientServerOperation to ArgumentDataPrototype in the target-role "argument"

- Remove source multiplicity 0..* from association from ClientServerOperation to ApplicationError in the target-role "possibleError"

- Remove source role "event" from dependency from OperationInvokedEvent to ClientServerOperation in the target-role "operation"

- Remove source multiplicity 1 from the association from ClientComSpec to ClientServerOperation in the target-role "operation"

- Remove source multiplicity 0..* from the association from ClientServerAnnotation to ClientServerOperation in the target-role "operation"

- Remove source multiplicity 1 from the association from ServerComSpec to ClientServerOperation in the target-role "operation"

- Remove source multiplicity * from the association from SwcImplementation to SwcInternalBehavior in the target-role "behavior"

- Remove source role "runnable" from aggregation from RunnableEntity to AsynchronousServerCallPoint in the target-role "asynchronousServerCallPoint"

- Remove source role "runnable" from aggregation from RunnableEntity to WaitPoint in the target-role "waitPoint"

- Remove source role "runnable" from aggregation from RunnableEntity to ModeSwitchPoint in the target-role "modeSwitchPoint"

- Remove source role "runnable" from aggregation from RunnableEntity to ServerCallPoint in the target-role "serverCallPoint"

- Remove source role "runnable" from aggregation from RunnableEntity to ExternalTriggeringPoint in the target-role "externalTriggeringPoint"

- Remove source role "event" and source p´multiplicity 0..1 from association from ModeSwitchedAckEvent to ModeSwitchPoint in the role "eventSource"

- Remove source multiplicity 1 from association from AsynchronousServerCallResultPoint to AsynchronousServerCallPoint in the target-role "asynchronousServerCallPoint", add constraint:

[constr_xxx1] Reference from AsynchronousServerCallResultPoint to AsynchronousServerCallPoint
In the context of a RunnabelEntity, a given AsynchronousServerCallPoint shall only be referenced by one AsynchronousServerCallResultPoint

- Remove source role "event" and source multiplicity 1 for association from AsynchronousServerCallReturnsEvent to AsynchronousServerCallResultPoint in the role "eventSource", add constraint:

[constr_xxx2] Reference from AsynchronousServerCallReturnsEvent to AsynchronousServerCallResultPoint
In the context of a RunnabelEntity, a given AsynchronousServerCallResultPoint shall only be referenced by one AsynchronousServerCallReturnsEvent

- Remove source role "event" and source multiplicity 1..* from association from InternalTriggerOccurredEvent to AsynchronousServerCallResultPoint in the target-role "eventSource"

- Remove source multiplicity * from association from WaitPoint to RTEEvent in the target-role "trigger"
–Last change on issue 77072 comment 10–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.26 Specification Item constr_1521

Trace References:

none

Content:

In the context of a RunnableEntity, a given AsynchronousServerCallResultPoint shall only be referenced by one AsynchronousServerCallReturnsEvent in the role Asynchronous ServerCallReturnsEvent.eventSource.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77072: Remove source multiplicities and source roles from the model

  Problem description:

  Remove all source multiplicities from the model.

  Agreed solution:

  Please implement in 80_Common and merge with the CP and AP branches.

  TPS_SystemTemplate:

  - the source multiplicity of the CommunicationController->CommunicationConnector association shall be removed (*).
  - the source multiplicity of the ISignal->SystemSignal association shall be removed (1..*).
  - the source multiplicity of the ISignalGroup->SystemSignalGroup association shall be removed (1..*).

- the source multiplicity of the FrameTriggering->Frame association shall be removed (1..*)
- the source multiplicity of the ISignalTriggering->ISignal association shall be removed (0..*)
- the source multiplicity of the ISignalIPduGroup->ISignalIPduGroup associationshall be removed (0..1).
- the source multiplicity of the PduTriggering->Pdu association shall be removed (1..*).
- the source multiplicity of the EcuInstance->ISignalIPduGroup association shall be removed (1).
- the source multiplicity of the NmNode->NmEcu association shall be removed (1..*).
- the source multiplicity of the CanNmClusterCoupling->CanNmCluster association shall be removed (0..1).
- the source multiplicity of the FlexrayNmClusterCoupling->FlexrayNmCluster association shall be removed (0..1).
- the source multiplicity of the UdpNmClusterCoupling->UdpNmCluster association shall be removed (0..1).
- the source multiplicity of the EcuResourceEstimation->EcuInstance association shall be removed (*)
- the source multiplicity of the EcuResourceEstimation->SwcToEcuMapping association shall be removed (*)
- the source multiplicity of the SignalPathConstraint->DocumentationBlock association shall be removed (0..1)
==
ECU Configuration specification:
- the source multiplicity of the EcucModuleConfigurationValues -> EcucContainerValue aggregation shall be removed (0..1)
- the source multiplicity of the EcucContainerValue -> EcucContainerValue aggregation shall be removed (0..1)
==
Ecu Resource Template:
- the source multiplicity of the HwElement -> HwElement association shall be removed (0..1)

==
Generic Structure Template:
- the source multiplicity of the RefConditional -> VariationPoint aggregation shall be removed (0)
==
Basic Software Module Description:

- the source multiplicity of the BswAsynchronousServerCallResultPoint -> BswAsynchronousServerCallPoint association shall be removed (1)

- the source multiplicity of the BswAsynchronousServerCallReturnsEvent -> BswAsynchronousServerCallResultPoint association shall be removed (1)
- the source multiplicity of the BswInternalTriggerOccurredEvent -> BswInternalTriggeringPoint association shall be removed (1..*)
===
SWCT:

- Remove source multiplicity 0..* from dependency from SwcModeSwitchEvent to ModeDeclaration in the target-role "mode"

- Remove source multiplicity 0..* from dependency from ModeAccessPoint to ModeDeclarationGroupPrototype in the target-role "modeGroup"

- Remove source multiplicity 1 and source role "interface" from aggregation from ModeSwitchInterface to ModeDeclarationGroupPrototype in the target-role "modeGroup", no constraint required because there is no reasonable multiplicity for aggregations other than 1 or 0..1.

- Remove source multiplicity 0..* from association from ModePortAnnotation to ModeDeclarationGroupPrototype in the target-role "modeGroup"

- Remove source multiplicity 0..* from dependency from ModeSwitchPoint to ModeDeclarationGroupPrototype in the target-role "modeGroup"

- Remove source multiplicity 0..* from association from TriggerPortAnnotation to Trigger in the target-role "trigger"

- Remove source role "event" from association from ExternalTriggerOccurredEvent to Trigger in the target-role "trigger"

- Remove source multiplicity 0..1 from association from ObdRatioServiceNeeds to DiagnosticEventNeeds in the target-role "rateBasedMonitoredEvent", add constraint:

[constr_xxxx] Semantics of ObdRatioServiceNeeds.rateBasedMonitoredEvent
In the context of an SwcServiceDependency, each DiagnostivEventNeeds referenced in the role rateBasedMonitoredEvent shall only be referenced by at most a single ObdRatioServiceNeeds.

- Remove source role "interface" from aggregation from NvDataInterface to VariableDataPrototype in the target-role "nvData"

- Remove source role "interface" and multiplicity 1 from aggregation from SenderReceiverInterface to VariableDataPrototype in the target-role "nvData"

- Remove source multiplicity 0..* from association from NvDataPortAnnotation to VariableDataPrototype in the target-role "variable"

- Remove source multiplicity 0..1 from association from SenderReceiverAnnotation to VariableDataPrototype in the target-role "dataElement", no constraint needed because annotations represent such a weak semantics

- Remove source role "event" from dependency from DataReceiveErrorEvent to VariableDataPrototype in the target-role "data"

- Remove source role "event" from dependency from DataReceivedEvent to VariableDataPrototype in the target-role "data"

- Remove source multiplicity 0..* from association from AssemblySwConnector to AbstractProvidedPortPrototype in the target-role "provider"

- Remove source multiplicity 0..* from association from AssemblySwConnector to AbstractRequiredPortPrototype in the target-role "requester"

- Remove source multiplicity * from association from SwComponentPrototype to SwComponentType in the target-role "type"

- Remove source role "component" from aggregation from SwComponentType to PortPrototype in the target-role "port"

- Remove source role "pPort" and multiplicity * from association from PPortPrototype to PortInterface in the target-role "providedInterface"

- Remove source multiplicity 0..* from dependency from DelegationSwConnector to PortPrototype in the target-role "innerPort"

- Remove source multiplicity 0..1 from association from PortAPIOption to PortPrototype in the target-role "port", no constraint needed because annotations represent such a weak semantics

- Remove source multiplicity 0..* from association from DelegationSwConnector to PortPrototype in the target-role "outerPort"

- Remove source role "rPort" and multiplicity * from association from RPortPrototype to PortInterface in the target-role "requiredInterface"

- Remove source multiplicity 0..1 from aggregation from ModeSwitchPoint to

PModeGroupInAtomicSwcInstanceRef in the target-role "modeGroup"

- Remove source multiplicity 1 and source role "event" from aggregation from OperationInvokedEvent to POperationInAtomicSwcInstanceRef in the target-role "operation"

- Remove source multiplicity 1..* and source role "event" from aggregation from ExternalTriggerOccurredEvent to RTriggerInAtomicSwcInstanceRef in the target-role "trigger"

- Remove source multiplicity 0..1 and source role "event" from aggregation from DataReceiveErrorEvent to RVariableInAtomicSwcInstanceRef in the target-role "data"

- Remove source multiplicity * and source role "event" from aggregation from DataReceivedEvent to RVariableInAtomicSwcInstanceRef in the target-role "data"

- Remove source role "composition" from aggregation from CompositionSwComponentType to SwConnector in the target-role "connector"

- Remove source multiplicity 1 and source role "interface" from aggregation from ClientServerInterface to ClientServerOperation in the target-role "operation"

- Remove source multiplicity 1 from aggregation from ClientServerOperation to ArgumentDataPrototype in the target-role "argument"

- Remove source multiplicity 0..* from association from ClientServerOperation to ApplicationError in the target-role "possibleError"

- Remove source role "event" from dependency from OperationInvokedEvent to ClientServerOperation in the target-role "operation"

- Remove source multiplicity 1 from the association from ClientComSpec to ClientServerOperation in the target-role "operation"

- Remove source multiplicity 0..* from the association from ClientServerAnnotation to ClientServerOperation in the target-role "operation"

- Remove source multiplicity 1 from the association from ServerComSpec to ClientServerOperation in the target-role "operation"

- Remove source multiplicity * from the association from SwcImplementation to SwcInternalBehavior in the target-role "behavior"

- Remove source role "runnable" from aggregation from RunnableEntity to AsynchronousServerCallPoint in the target-role "asynchronousServerCallPoint"

- Remove source role "runnable" from aggregation from RunnableEntity to WaitPoint in the target-role "waitPoint"

- Remove source role "runnable" from aggregation from RunnableEntity to ModeSwitchPoint in the target-role "modeSwitchPoint"

- Remove source role "runnable" from aggregation from RunnableEntity to ServerCallPoint in the target-role "serverCallPoint"

- Remove source role "runnable" from aggregation from RunnableEntity to ExternalTriggeringPoint in the target-role "externalTriggeringPoint"

- Remove source role "event" and source p´multiplicity 0..1 from association from ModeSwitchedAckEvent to ModeSwitchPoint in the role "eventSource"

- Remove source multiplicity 1 from association from AsynchronousServerCallResultPoint to AsynchronousServerCallPoint in the target-role "asynchronousServerCallPoint", add constraint:

[constr_xxx1] Reference from AsynchronousServerCallResultPoint to AsynchronousServerCallPoint
In the context of a RunnabelEntity, a given AsynchronousServerCallPoint shall only be referenced by one AsynchronousServerCallResultPoint

- Remove source role "event" and source multiplicity 1 for association from AsynchronousServerCallReturnsEvent to AsynchronousServerCallResultPoint in the role "eventSource", add constraint:

[constr_xxx2] Reference from AsynchronousServerCallReturnsEvent to AsynchronousServerCallResultPoint
In the context of a RunnabelEntity, a given AsynchronousServerCallResultPoint shall only be referenced by one AsynchronousServerCallReturnsEvent

- Remove source role "event" and source multiplicity 1..* from association from InternalTriggerOccurredEvent to AsynchronousServerCallResultPoint in the target-role "eventSource"

- Remove source multiplicity * from association from WaitPoint to RTEEvent in

the target-role "trigger"
–Last change on issue 77072 comment 10–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.27 Specification Item constr_1523

Trace References:

none

Content:

An OperationInvokedEvent shall not have a reference to a ModeDeclaration in the role RTEEvent.disabledMode.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76989: Missing constraints for mode disabling dependencies and OperationInvokedEvents / BswOperationInvokedEvents

  Problem description:

  The SWS RTE excludes
  the combination of mode disabling dependencies and OperationInvokedEvents ([SWS_Rte_02706])

  Reading the rational, the same combination shall be excluded for BswOperationInvokedEvents

  Agreed solution:

  SWC-T
  [constr_xxxx] No mode disabling for OperationInvokedEvents d An OperationInvokedEvent shall not have a reference to a ModeDeclaration in the role disabledMode. c()

  Rationale
  The RTE does not support the disabling of server Runnables by modes. Instead, the server shall respond with an explicit error code if the execution of the server operation is not possible in specific side conditions.

BSWMD-T
[constr_yyyy] No mode disabling for BswOperationInvokedEvents d An BswOperationInvokedEvents shall not have a reference to a ModeDeclaration in the role disabledInMode. c()

SWS RTE:
adjust
[SWS_Rte_02706] The RTE shall reject the configurations violating constr_xxxx
add
[SWS_Rte_nnnn1] d The RTE shall reject the configurations violating constr_yyyy
–Last change on issue 76989 comment 3–

BW-C-Level:

Application	Specification	Bus
4	4	1

1.28 Specification Item constr_1538

Trace References:

none

Content:

The reference ReceiverComSpec.ReceiverComSpec.dataElementshall not refer to an ArgumentDataPrototype or ParameterDataPrototype.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77984: Change destination of ReceiverComSpec and SenderComSpec dataElement references to AutosarDataPrototype

  Problem description:

  This RfC is a result of the Discussions in the Adaptive Platform.
  In the current metamodel for 17-10 the ReceiverComSpec refers a VariableDataPrototype in the role dataElement (reference also available in CP) and in addition it refers the Field in the role fieldNotifier (reference is restricted to AP). The same is valid for the SenderComSpec.

  We introduced constr_3381 and constr_3382 that defines that the ReceiverComSpec (SenderComSpec) shall not reference a dataElement and a fieldNotifier at the same time.

Another possible solution would be to change the destination of the dataElement reference to the AutosarDataPrototype since the VariableDataPrototype and the Field are both specializations of the AutosarDataPrototype. With this solution we would need to restrict the reference with a new constraint in the Software Component Template to the VariableDataPrototype and in the Manifest to Field and VariableDataPrototype.

The Software Component Group shall discuss in the next meeting which solution is preferred.

Agreed solution:

change the target of the ReceiverComSpec.dataElement and SenderComSpec.dataElement from VariableDataPrototype to AutosarDataPrototype.

constr_xxxx
The references ReceiverComSpec.dataElement and SenderComSpec.dataElement shall not point to an ArgumentDataPrototype or ParameterDataPrototype.

Set the multiplicity of ClientComSpec.operation and ServerComSpec.operation to 0..1.

constr_xxx1
The reference ClientComSpec.operation shall exist if the AbstractRequiredPortPrototype that owns the ClientComSpec is typed by a ClientServerInterface

constr_xxx2
The reference ServerComSpec.operation shall exist if the AbstractProvidedPortPrototype that owns the ServerComSpec is typed by a ClientServerInterface

Note: on the AUTOSAR adaptive platform these ComSpecs can also be used in the context of PortInterfaces that are not available on the classic platform. This is the motivation for the existence of constr_xxx1 and constr_xxx2.
–Last change on issue 77984 comment 5–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.29 Specification Item constr_1539

Trace References:

none

Content:

The reference SenderComSpec.SenderComSpec.dataElementshall not refer to an ArgumentDataPrototype or ParameterDataPrototype.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77984: Change destination of ReceiverComSpec and SenderComSpec dataElement references to AutosarDataPrototype

  Problem description:

  This RfC is a result of the Discussions in the Adaptive Platform.
  In the current metamodel for 17-10 the ReceiverComSpec refers a VariableDataPrototype in the role dataElement (reference also available in CP) and in addition it refers the Field in the role fieldNotifier (reference is restricted to AP). The same is valid for the SenderComSpec.

  We introduced constr_3381 and constr_3382 that defines that the ReceiverComSpec (SenderComSpec) shall not reference a dataElement and a fieldNotifier at the same time.

  Another possible solution would be to change the destination of the dataElement reference to the AutosarDataPrototype since the VariableDataPrototype and the Field are both specializations of the AutosarDataPrototype. With this solution we would need to restrict the reference with a new constraint in the Software Component Template to the VariableDataPrototype and in the Manifest to Field and VariableDataPrototype.

  The Software Component Group shall discuss in the next meeting which solution is preferred.

  Agreed solution:

  change the target of the ReceiverComSpec.dataElement and SenderComSpec.dataElement from VariableDataPrototype to AutosarDataPrototype.

  constr_xxxx
  The references ReceiverComSpec.dataElement and SenderComSpec.dataElement shall not point to an ArgumentDataPrototype or ParameterDataPrototype.

  Set the multiplicity of ClientComSpec.operation and ServerComSpec.operation to 0..1.

constr_xxx1
The reference ClientComSpec.operation shall exist if the AbstractRequiredPortPrototype that owns the ClientComSpec is typed by a ClientServerInterface

constr_xxx2
The reference ServerComSpec.operation shall exist if the AbstractProvidedPortPrototype that owns the ServerComSpec is typed by a ClientServerInterface

Note: on the AUTOSAR adaptive platform these ComSpecs can also be used in the context of PortInterfaces that are not available on the classic platform. This is the motivation for the existence of constr_xxx1 and constr_xxx2.
–Last change on issue 77984 comment 5–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.30 Specification Item constr_1540

Trace References:

none

Content:

The reference ClientComSpec.ClientComSpec.operationshall exist if the AbstractRequiredPortPrototype that owns the ClientComSpec is typed by a ClientServerInterface.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77984: Change destination of ReceiverComSpec and SenderComSpec dataElement references to AutosarDataPrototype

  Problem description:

  This RfC is a result of the Discussions in the Adaptive Platform.
  In the current metamodel for 17-10 the ReceiverComSpec refers a VariableDataPrototype in the role dataElement (reference also available in CP) and in addition it refers the Field in the role fieldNotifier (reference is restricted to AP). The same is valid for the SenderComSpec.

  We introduced constr_3381 and constr_3382 that defines that the ReceiverComSpec (SenderComSpec) shall not reference a dataElement and a fieldNotifier at the same time.

Another possible solution would be to change the destination of the dataElement reference to the AutosarDataPrototype since the VariableDataPrototype and the Field are both specializations of the AutosarDataPrototype. With this solution we would need to restrict the reference with a new constraint in the Software Component Template to the VariableDataPrototype and in the Manifest to Field and VariableDataPrototype.

The Software Component Group shall discuss in the next meeting which solution is preferred.

Agreed solution:

change the target of the ReceiverComSpec.dataElement and SenderComSpec.dataElement from VariableDataPrototype to AutosarDataPrototype.

constr_xxxx
The references ReceiverComSpec.dataElement and SenderComSpec.dataElement shall not point to an ArgumentDataPrototype or ParameterDataPrototype.

Set the multiplicity of ClientComSpec.operation and ServerComSpec.operation to 0..1.

constr_xxx1
The reference ClientComSpec.operation shall exist if the AbstractRequiredPortPrototype that owns the ClientComSpec is typed by a ClientServerInterface

constr_xxx2
The reference ServerComSpec.operation shall exist if the AbstractProvidedPortPrototype that owns the ServerComSpec is typed by a ClientServerInterface

Note: on the AUTOSAR adaptive platform these ComSpecs can also be used in the context of PortInterfaces that are not available on the classic platform. This is the motivation for the existence of constr_xxx1 and constr_xxx2.
–Last change on issue 77984 comment 5–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.31 Specification Item constr_1541

Trace References:

none

Content:

The reference ServerComSpec.ServerComSpec.operationshall exist if the AbstractProvidedPortPrototype that owns the ServerComSpec is typed by a ClientServerInterface.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77984: Change destination of ReceiverComSpec and SenderComSpec dataElement references to AutosarDataPrototype

  Problem description:

  This RfC is a result of the Discussions in the Adaptive Platform.
  In the current metamodel for 17-10 the ReceiverComSpec refers a VariableDataPrototype in the role dataElement (reference also available in CP) and in addition it refers the Field in the role fieldNotifier (reference is restricted to AP). The same is valid for the SenderComSpec.

  We introduced constr_3381 and constr_3382 that defines that the ReceiverComSpec (SenderComSpec) shall not reference a dataElement and a fieldNotifier at the same time.

  Another possible solution would be to change the destination of the dataElement reference to the AutosarDataPrototype since the VariableDataPrototype and the Field are both specializations of the AutosarDataPrototype. With this solution we would need to restrict the reference with a new constraint in the Software Component Template to the VariableDataPrototype and in the Manifest to Field and VariableDataPrototype.

  The Software Component Group shall discuss in the next meeting which solution is preferred.

  Agreed solution:

  change the target of the ReceiverComSpec.dataElement and SenderComSpec.dataElement from VariableDataPrototype to AutosarDataPrototype.

  constr_xxxx
  The references ReceiverComSpec.dataElement and SenderComSpec.dataElement

shall not point to an ArgumentDataPrototype or ParameterDataPrototype.

Set the multiplicity of ClientComSpec.operation and ServerComSpec.operation to 0..1.

constr_xxx1
The reference ClientComSpec.operation shall exist if the AbstractRequiredPortPrototype that owns the ClientComSpec is typed by a ClientServerInterface

constr_xxx2
The reference ServerComSpec.operation shall exist if the AbstractProvidedPortPrototype that owns the ServerComSpec is typed by a ClientServerInterface

Note: on the AUTOSAR adaptive platform these ComSpecs can also be used in the context of PortInterfaces that are not available on the classic platform. This is the motivation for the existence of constr_xxx1 and constr_xxx2.
–Last change on issue 77984 comment 5–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.32 Specification Item constr_1544

Trace References:

none

Content:

The standardized values and multiplicities within the model of an SwAxisGeneric according to TPS_SWCT_01479 and TPS_SWCT_01480 are documented in Table [REF table_3a_ModelingOfSwAxisGeneric].

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77620: Clarifiy description of FIX_AXIS

  Problem description:

  The AUTOSAR Meta Model is prepared for the description of FIX_AXIS but the SWC-T does not provide further information how to use it.

  Agreed solution:

SWCT
====

Create new chapter 5.4.4.1 Specification of fix axes

In most cases the axes of a curve or map are accessible to a calibration software and it is possible to calibrate axes points and their corresponding values. There are cases, however, where axes are intentionally declared as fix and where no intention exists to change the properties of the axis ever[fn1].

These axes are also known as fix axes.

[TPS_SWCT_0xxx] Value of category for fix axis
A fix axis shall be modeled as an SwCalprmAxis with attribute category set to the value FIX_AXIS.

[TPS_SWCT_0xxx1] Sub-categories of fix axes
There are different sub-categories of fix axes:
* Fix axis where the distance between axis points can be computed according to a standardized algorithm. In this case, fix axes of arbitrary length can be described by feeding three arguments defined in the context of the axis description into the axis algorithm. Consequently, the memory footprint of different fix axis of this category is literally identical, independently of the number of axis points.
The following variations exist:
- Subcategory PAR, i.e. category = FIX_AXIS_PAR: the axis is created out of a starting value and a shift that creates further axis points as using a power-of-two algorithm. The details can be found in [27].
- Subcategory PAR_DIST, i.e. category = FIX_AXIS_PAR_DIST: the axis is created out of a starting value and an offset that adds further axis points with the distance given by offset. The details can be found in [27].
* Fix axis where the axis points are defined as a list of values directly in the axis definition. This variety boils down to
- Subcategory PAR_LIST, i.e. category = FIX_AXIS_PAR_LIST: the axis is created out of a list of numerical values that represent the axis points. The details can be found in [27].
These values of category shall be used for SwAxisType.

In other words, since the values that define a fix axis are modeled directly in the axis model there is no use case for assigning an initial value to a fix axis.

However, the calibration software may have still have access to axis points and values of the fix axis if these properties are specified in an A2L file. For this purpose McDataInstance needs to be set up properly. The details are explained in [7].

The modeling of of an axis of category FIX_AXIS_PAR is sketched in the following example model.

<insert attachment 4552>

The modeling of of an axis of category FIX_AXIS_PAR_DIST is sketched in the following example model.

<insert attachment 4553>

The modeling of of an axis of category FIX_AXIS_PAR_LIST is sketched in the following example model.

<insert attachment 4555>

Please note that the axis points and values of a fix axis are defined in the definition of the fix axis itself and therefore any initial value asigned to a fix axis would be ignored anyway. This might lead to confusion such that the initial value does not make it into the software. In order to avoid such confusion AUTOSAR does not support the definition of an initial value for a fix axis. This regulation is reflected in the existence of constr_xxx1.

[constr_0xxx1] No initialization for fix axis
an ApplicationValueSpecifciation taken to initialize an ApplicationPrimitiveDataType that contains a fix axis shall not contain initial values for the axis index of the fix axis inside the ApplicationPrimitiveDataType.

Add table:

category of swAxisType category of SwGenericAxisParamType Multiplicity
FIX_AXIS_PAR OFFSET 1
SHIFT 1
FIX_AXIS_PAR_DIST OFFSET 1
DISTANCE 1
FIX_AXIS_PAR_LIST LIST 1

Add constraint that references the table.

Changes in the meta-model:

* Remove the mention of FIX_AXIS from the description of RuleBasedAxisCont.category (because fix axis are not initialized by means of a ValueSpecifiction)

* Remove the mention of FIX_AXIS from the description of SwAxisCont.category (because fix axis are not initialized by means of a ValueSpecifiction)

* make SwAxisGenericParam.vf ordered.

[fn1]: Typically, a calibration software does not have the ability to manipulate (or even inspect) the axis' properties by inspecting the ECU's memory.

[27]: ASAM MCD 2MC ASAP2 Interface Specification

[7]: Basic Software Module Description Template

GenBlprSupplement
=================

3.2.3 Record Layout: FixIntCur

This chapter describes the record layout for a curve with fixed axis points. Fixed axis exist in three categories: FIX_AXIS_PAR, FIX_AXIS_PAR_DIST and FIX_AXIS_PAR_LIST, see [TPS_SWCT_0xxx1].

The number of sampling points (Nx), the Offset, the shift and the distance values are represented in the following chapters by these logical views:

<insert figure with three representations>

These values are not defined inside SwRecordLayout(s) with fixed axis points.

Logical view:

The figure 3.9 illustrates the logical view of the SwRecordLayout FixIntCur. The SwRecordLayoutGroup with the shortLabel Val is shown in the lower part. Its elements are indexed by virtual [AXIS 1] which is fixed and of category FIX_AXIS_PAR and not defined inside this SwRecordLayout.

3.3.6 Record Layout: FixIntMap

This chapter describes the record layout for a map with fixed axis points. Fixed axis exist in three categories: FIX_AXIS_PAR, FIX_AXIS_PAR_DIST and FIX_AXIS_PAR_LIST, see [TPS_SWCT_xxxxx].

The number of sampling points (Nx, Ny), the Offset, the shift and the distance values are represented in the following chapters by these logical views:

<insert figure with three representations>

These values are not defined inside SwRecordLayout(s) with fixed axis points.

Logical view:
The figure 3.27 illustrates the logical view of the SwRecordLayout FixIntMap. The SwRecordLayoutGroup with the shortLabel Val is shown in the lower part. Its elements are indexed by [AXIS 2, AXIS 1] from value (AXIS 2: = 1, AXIS 1: = 1) to value (AXIS 2: = -1, AXIS 1: = -1) there -1 gives the last value.
–Last change on issue 77620 comment 26–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.33 Specification Item constr_1545

Trace References:

none

Content:

An ApplicationValueSpecification taken to initialize an ApplicationPrimitiveDataType that contains a fix axis shall not contain initial values for the axis index of the fix axis inside the ApplicationPrimitiveDataType.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77620: Clarifiy description of FIX_AXIS

  Problem description:

  The AUTOSAR Meta Model is prepared for the description of FIX_AXIS but the SWC-T does not provide further information how to use it.

  Agreed solution:

SWCT
====

Create new chapter 5.4.4.1 Specification of fix axes

In most cases the axes of a curve or map are accessible to a calibration software and it is possible to calibrate axes points and their corresponding values. There are cases, however, where axes are intentionally declared as fix and where no intention exists to change the properties of the axis ever[fn1].

These axes are also known as fix axes.

[TPS_SWCT_0xxx] Value of category for fix axis
A fix axis shall be modeled as an SwCalprmAxis with attribute category set to the value FIX_AXIS.

[TPS_SWCT_0xxx1] Sub-categories of fix axes
There are different sub-categories of fix axes:
* Fix axis where the distance between axis points can be computed according to a standardized algorithm. In this case, fix axes of arbitrary length can be described by feeding three arguments defined in the context of the axis description into the axis algorithm. Consequently, the memory footprint of different fix axis of this category is literally identical, independently of the number of axis points.
The following variations exist:
- Subcategory PAR, i.e. category = FIX_AXIS_PAR: the axis is created out of a starting value and a shift that creates further axis points as using a power-of-two algorithm. The details can be found in [27].
- Subcategory PAR_DIST, i.e. category = FIX_AXIS_PAR_DIST: the axis is created out of a starting value and an offset that adds further axis points with the distance given by offset. The details can be found in [27].
* Fix axis where the axis points are defined as a list of values directly in the axis definition. This variety boils down to
- Subcategory PAR_LIST, i.e. category = FIX_AXIS_PAR_LIST: the axis is created out of a list of numerical values that represent the axis points. The details can be found in [27].
These values of category shall be used for SwAxisType.

In other words, since the values that define a fix axis are modeled directly in the axis model there is no use case for assigning an initial value to a fix axis.

However, the calibration software may have still have access to axis points and values of the fix axis if these properties are specified in an A2L file. For this purpose McDataInstance needs to be set up properly. The details are explained in [7].

The modeling of of an axis of category FIX_AXIS_PAR is sketched in the following example model.

<insert attachment 4552>

The modeling of of an axis of category FIX_AXIS_PAR_DIST is sketched in the following example model.

<insert attachment 4553>

The modeling of of an axis of category FIX_AXIS_PAR_LIST is sketched in the following example model.

<insert attachment 4555>

Please note that the axis points and values of a fix axis are defined in the definition of the fix axis itself and therefore any initial value asigned to a fix axis would be ignored anyway. This might lead to confusion such that the initial value does not make it into the software. In order to avoid such confusion AUTOSAR does not support the definition of an initial value for a fix axis. This regulation is reflected in the existence of constr_xxx1.

[constr_0xxx1] No initialization for fix axis
an ApplicationValueSpecifciation taken to initialize an ApplicationPrimitiveDataType that contains a fix axis shall not contain initial values for the axis index of the fix axis inside the ApplicationPrimitiveDataType.

Add table:

category of swAxisType category of SwGenericAxisParamType Multiplicity
FIX_AXIS_PAR OFFSET 1
SHIFT 1
FIX_AXIS_PAR_DIST OFFSET 1
DISTANCE 1
FIX_AXIS_PAR_LIST LIST 1

Add constraint that references the table.

Changes in the meta-model:

* Remove the mention of FIX_AXIS from the description of RuleBasedAxisCont.category (because fix axis are not initialized by means of a ValueSpecifiction)

* Remove the mention of FIX_AXIS from the description of SwAxisCont.category (because fix axis are not initialized by means of a ValueSpecifiction)

* make SwAxisGenericParam.vf ordered.

[fn1]: Typically, a calibration software does not have the ability to manipulate (or even inspect) the axis' properties by inspecting the ECU's memory.

[27]: ASAM MCD 2MC ASAP2 Interface Specification

[7]: Basic Software Module Description Template

GenBlprSupplement
=================

3.2.3 Record Layout: FixIntCur

This chapter describes the record layout for a curve with fixed axis points. Fixed axis exist in three categories: FIX_AXIS_PAR, FIX_AXIS_PAR_DIST and FIX_AXIS_PAR_LIST, see [TPS_SWCT_0xxx1].

The number of sampling points (Nx), the Offset, the shift and the distance values are represented in the following chapters by these logical views:

<insert figure with three representations>

These values are not defined inside SwRecordLayout(s) with fixed axis points.

Logical view:

The figure 3.9 illustrates the logical view of the SwRecordLayout FixIntCur. The SwRecordLayoutGroup with the shortLabel Val is shown in the lower part. Its elements are indexed by virtual [AXIS 1] which is fixed and of category FIX_AXIS_PAR and not defined inside this SwRecordLayout.

3.3.6 Record Layout: FixIntMap

This chapter describes the record layout for a map with fixed axis points. Fixed axis exist in three categories: FIX_AXIS_PAR, FIX_AXIS_PAR_DIST and FIX_AXIS_PAR_LIST, see [TPS_SWCT_xxxxx].

The number of sampling points (Nx, Ny), the Offset, the shift and the distance values are represented in the following chapters by these logical views:

<insert figure with three representations>

These values are not defined inside SwRecordLayout(s) with fixed axis points.

Logical view:
The figure 3.27 illustrates the logical view of the SwRecordLayout FixIntMap. The SwRecordLayoutGroup with the shortLabel Val is shown in the lower part. Its elements are indexed by [AXIS 2, AXIS 1] from value (AXIS 2: = 1, AXIS 1: = 1) to value (AXIS 2: = -1, AXIS 1: = -1) there -1 gives the last value.
–Last change on issue 77620 comment 26–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.34 Specification Item constr_2027

Trace References:

none

Content:

A PortPrototype that is referenced by a SwcServiceDependency via SwcServiceDependency.assignedPort or via SwcServiceDependency.assignedData shall be typed by a Port Interface that has PortInterface.isService set to true.

This rule does not apply to PortPrototypes used in the context of NV data management, i.e. for connections between an ApplicationSwComponentType and an NvBlockSwComponentType. referenced by a RoleBasedPortAssignment where the attribute RoleBased PortAssignment.role is set to any of the following values:

- NvMService
- NvMNotifyJobFinished
- NvMNotifyInitBlock

- NvMAdmin
- NvMMirror
- NvDataPort

Furthermore, the rule does not apply to the case described in TPS_SWCT_01579, TPS_SWCT_01580, as well as TPS_SWCT_01572.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #74121: SwcServiceDependency for service ports with isService set to false

  Problem description:

  In constr_2027 it is unclear what exactly "used in the context of NV data management" means. In the second part of the sentence it tries to clarify that with "for connections between an ApplicationSwComponentType and an NvBlockSwComponentType".
  However, I think that a constraint should not depend on the connection of a port. Otherwise, an unconnected port would violate that rule.

  precise this constraint e.g. by saying that
  "... This rule does not apply to PortPrototypes where the attribute role of a RoleBasedPortAssignment is set to 'NvDataPort'".
  –Last change on issue 74121 comment 8–

  Agreed solution:

  In constraint 2027, replace the "used in the context of NV data management" by

  This rule does not apply to PortPrototypes where the attribute role of a RoleBasedPortAssignment is set to any of these roles:

  * NvMService
  * NvMNotifyJobFinished
  * NvMNotifyInitBlock
  * NvMAdmin
  * NvMMirror
  * NvDataPort

  Add a reference to TPS_SWCT_01579 within TPS_SWCT_02003.

  Add a further case to constr_2027: described by TPS_SWCT_01579 and TPS_SWCT_01580 (parasitic data access)
  Add a further case to cinstr_2027: described by TPS_SWCT_01572 (Inteaction of

ApplicationSwComponent with both the BswM and another ApplicationSwComponent)
–Last change on issue 74121 comment 1–

BW-C-Level:

Application	Specification	Bus
1	1	1

- RfC #77507: Add TPS_SWCT_01580 to the list of exceptions in constr_2027

  Problem description:

  constr_1580 makes some exceptions for the existence of SwcServiceDependencies. One exception (TPS_SWCT_01579) applies to the access of dataElements directly from the Dcm. The Dem, however, has the same use case but is not considered appropriately. Therefore, we should add TPS_SWCT_01580 to the list of exceptions in constr_2027

  Agreed solution:

  Rephrase constr_2027:

  [constr_2027] SwcServiceDependency shall be defined for service ports only d
  A PortPrototype that is referenced by a SwcServiceDependency via assignedPort or via assignedData shall be typed by a PortInterface that has isService set to true.
  This rule does not apply to PortPrototypes referenced by a RoleBasedPortAssignment where the attribute role is set to any of the following values:
  NvMService
  NvMNotifyJobFinished
  NvMNotifyInitBlock
  NvMAdmin
  NvMMirror
  NvDataPort
  Furthermore, the rule does not apply to the case described in [TPS_SWCT_01579], [TPS_SWCT_01580], as
  well as [TPS_SWCT_01572]. c()
  –Last change on issue 77507 comment 1–

  BW-C-Level:

Application	Specification	Bus
1	1	1

1.35 Specification Item constr_2043

Trace References:

none

Content:

The overriding SwDataDefProps.swImplPolicy attribute value of a ParameterDataPrototype in the role NvBlockDescriptor.romBlock shall be SwImplPolicyEnum.standard.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #74975: constr_2043 and constr_2046 in TPS_SWCT refer to wrong roles

  Problem description:

  Name: Ulrich Kiffmeier
  Phone: 05251 1638 747
  Role: WP-M Member

  The constraints 2043 and 2046 seem to be broken:

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

  ==> staticMemory is a VariableDataPrototype, not a ParameterDataPrototype. I think the constraint is about the swImplPolicy for a NvBlockDescriptor.romBlock?

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

  ==> Headline refers to the role constantMemory while text refers to the role sharedParameter.

  My proposed solution would be:
  [constr_2043] swImplPolicy for ParameterDataPrototype in the role *romBlock*
  The overriding swImplPolicy attribute value of a ParameterDataPrototype in the role

romBlock shall be standard.

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

Agreed solution:

Replacements for existing constraints:

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

[constr_2044] swImplPolicy for ParameterDataPrototype in the role sharedParameter
The overriding swImplPolicy attribute value of a ParameterDataPrototype in the role sharedParameter shall be standard, const

[constr_2045] swImplPolicy for ParameterDataPrototype in the role perInstanceParameter
The overriding swImplPolicy attribute value of a ParameterDataPrototype in the role perInstanceParameter shall be standard, const

[constr_2046] swImplPolicy for ParameterDataPrototype in the role constantMemory
The overriding swImplPolicy attribute value of a ParameterDataPrototype in the role constantMemory shall be standard, const or fixed.
–Last change on issue 74975 comment 4–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.36 Specification Item constr_2044

Trace References:

none

Content:

The overriding SwDataDefProps.swImplPolicy attribute value of a ParameterDataPrototype in the role SwcInternalBehavior.sharedParameter shall be SwImplPolicyEnum.standard, SwImplPolicyEnum.const.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #74975: constr_2043 and constr_2046 in TPS_SWCT refer to wrong roles

  Problem description:

  Name: Ulrich Kiffmeier
  Phone: 05251 1638 747
  Role: WP-M Member

  The constraints 2043 and 2046 seem to be broken:

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

  ==> staticMemory is a VariableDataPrototype, not a ParameterDataPrototype. I think the constraint is about the swImplPolicy for a NvBlockDescriptor.romBlock?

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

  ==> Headline refers to the role constantMemory while text refers to the role sharedParameter.

  My proposed solution would be:
  [constr_2043] swImplPolicy for ParameterDataPrototype in the role *romBlock*
  The overriding swImplPolicy attribute value of a ParameterDataPrototype in the role romBlock shall be standard.

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

Agreed solution:

Replacements for existing constraints:

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

[constr_2044] swImplPolicy for ParameterDataPrototype in the role sharedParameter
The overriding swImplPolicy attribute value of a ParameterDataPrototype in the role sharedParameter shall be standard, const

[constr_2045] swImplPolicy for ParameterDataPrototype in the role perInstanceParameter
The overriding swImplPolicy attribute value of a ParameterDataPrototype in the role perInstanceParameter shall be standard, const

[constr_2046] swImplPolicy for ParameterDataPrototype in the role constantMemory
The overriding swImplPolicy attribute value of a ParameterDataPrototype in the role constantMemory shall be standard, const or fixed.
–Last change on issue 74975 comment 4–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.37 Specification Item constr_2045

Trace References:

none

Content:

The overriding SwDataDefProps.swImplPolicy attribute value of a ParameterDataPrototype in the role SwcInternalBehavior.sharedperInstanceParameter shall be SwImplPolicyEnum.standard, SwImplPolicyEnum.const.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #74975: constr_2043 and constr_2046 in TPS_SWCT refer to wrong roles

Problem description:

Name: Ulrich Kiffmeier
Phone: 05251 1638 747
Role: WP-M Member

The constraints 2043 and 2046 seem to be broken:

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

==> staticMemory is a VariableDataPrototype, not a ParameterDataPrototype.
I think the constraint is about the swImplPolicy for a NvBlockDescriptor.romBlock?

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

==> Headline refers to the role constantMemory while text refers to the role sharedParameter.

My proposed solution would be:
[constr_2043] swImplPolicy for ParameterDataPrototype in the role *romBlock*
The overriding swImplPolicy attribute value of a ParameterDataPrototype in the role romBlock shall be standard.

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

Agreed solution:

Replacements for existing constraints:

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

[constr_2044] swImplPolicy for ParameterDataPrototype in the role sharedParameter
The overriding swImplPolicy attribute value of a ParameterDataPrototype in the role sharedParameter shall be standard, const

[constr_2045] swImplPolicy for ParameterDataPrototype in the role perInstanceParameter
The overriding swImplPolicy attribute value of a ParameterDataPrototype in the role perInstanceParameter shall be standard, const

[constr_2046] swImplPolicy for ParameterDataPrototype in the role constantMemory
The overriding swImplPolicy attribute value of a ParameterDataPrototype in the role constantMemory shall be standard, const or fixed.
–Last change on issue 74975 comment 4–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.38 Specification Item constr_2046

Trace References:

none

Content:

The overriding SwDataDefProps.swImplPolicy attribute value of a ParameterDataPrototype in the role SwcInternalBehavior.sharedParameter constantMemory shall be SwImpl PolicyEnum.standard, SwImplPolicyEnum.const or SwImplPolicyEnum.fixed.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #74975: constr_2043 and constr_2046 in TPS_SWCT refer to wrong roles

  Problem description:

  Name: Ulrich Kiffmeier
  Phone: 05251 1638 747
  Role: WP-M Member

The constraints 2043 and 2046 seem to be broken:

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

==> staticMemory is a VariableDataPrototype, not a ParameterDataPrototype. I think the constraint is about the swImplPolicy for a NvBlockDescriptor.romBlock?

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

==> Headline refers to the role constantMemory while text refers to the role sharedParameter.

My proposed solution would be:
[constr_2043] swImplPolicy for ParameterDataPrototype in the role *romBlock*
The overriding swImplPolicy attribute value of a ParameterDataPrototype in the role romBlock shall be standard.

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

Agreed solution:

Replacements for existing constraints:

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

[constr_2044] swImplPolicy for ParameterDataPrototype in the role sharedParameter
The overriding swImplPolicy attribute value of a ParameterDataPrototype in the role sharedParameter shall be standard, const

[constr_2045] swImplPolicy for ParameterDataPrototype in the role perInstan-

ceParameter
The overriding swImplPolicy attribute value of a ParameterDataPrototype in the role perInstanceParameter shall be standard, const

[constr_2046] swImplPolicy for ParameterDataPrototype in the role constantMemory
The overriding swImplPolicy attribute value of a ParameterDataPrototype in the role constantMemory shall be standard, const or fixed.
–Last change on issue 74975 comment 4–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.39 Specification Item TPS_SWCT_01193

Trace References:

none

Content:

In general, it is not required that the sum of all mappings between ApplicationDataType and ImplementationDataType in a given system form a 1:1 relation. Depending on the use case and on the scope, 1:n as well as n:1 mappings are possible :

- Several different ApplicationDataTypes may be mapped to the same Implementation DataType in the scope of a system, an ECU, or even a single InternalBehavior of an atomic software component.

  Of course, this requires that the different ApplicationDataTypes are used for different DataPrototypes and thus that the DataPrototypes are typed by them (and not by the ImplementationDataTypes). This allows to establish a more simple type system on the implementation level, than on the application model level.

- The same ApplicationDataTypes may be mapped to different ImplementationData Types for different ECUs. This scenario allows to chose the implementation data types according to the needs of specific ECUs.

- The same ApplicationDataTypes may be mapped to different ImplementationData Types even in the scope of a single ECU (more exactly speaking, a single RTE), but only for different AtomicSwComponentTypes (see constr_1004).

  This improves the portability of software components which were developed independently or are ported between ECUs.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77040: Nested constraint in TPS_SWCT_01193

  Problem description:

  TPS_SWCT_01193 contains the definition of constr_1004, which is not allowed due to the existence of constr_2565 (STDT).
  –Last change on issue 77040 comment 3–

  Agreed solution:

  SWCT
  ====

  Remove constr_1004 and the note below from the list of bullet points inside TPS_SWCT_01193 and place constr_1004 and the note immediately below TPS_SWCT_01193.

  Replace the following sentence inside TPS_SWCT_01193

  Depending on the use case and on the scope, 1:n as well as n:1 mappings are possible:

  with

  Depending on the use case and on the scope, 1:n as well as n:1 mappings are possible (constr_1004 applies):

  TOOL
  ====

  Let CheckDocumentSource complain about nested traces.
  –Last change on issue 77040 comment 4–

  BW-C-Level:

Application	Specification	Bus
1	1	1

1.40 Specification Item TPS_SWCT_01222

Trace References:

RS_SWCT_03221

Content:

This OSEK ISO 17356-4 specification states that “filtering is only used for messages that can be interpreted as C language unsigned integer types (characters, unsigned integers and enumerations).”

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #73564: References to OSEK

  Problem description:

  The OSEK web site is not available.
  References need to be updated.

  OSEK was pushed to ISO 17356
  ISO 17356-2: introduction?
  ISO 17356-3: OS
  ISO 17356-4: COM
  ISO 17356-5: NM (not sure we need to have such reference)
  ISO 17356-6: OIL (not sure we need to have such reference)

  I tried to select the documents where it would make sense to get a fix.

  Agreed solution:

  Replace references to
  * OSEK web site
  * the OSEK file name / version
  with an ISO reference (with ISO version)

  Check references to OSEK subsection (or avoid such references), to make sure that the section numbering in ISO is the same.

  SWS OS
  ======

  Change references in chapter 3.2.1 to ISO. Remove [16], [18], [19] and [20]

  Remove [22] from chapter 3.2.2

  Remove "OSEKtime OS [16] and the HIS Protected OSEK [22] are immature specifications that contain concepts necessary for AUTOSAR and satisfy specific application domains. It is the purpose of this document to identify these needs and to recommend the use of parts (or all) of these specifications as appropriate." from chapter 4.1.

Change "OSEK OS" to new ISO reference ("OSEK OS[..]" - many occurences, also in chapter 10)

Remove "So called hard and smooth synchronization from OSEKtime [16] are supported by this single unified concept in AUTOSAR OS. Smooth synchronization may be emulated by setting the small adjustment values on the final expiry point. Hard synchronization may be emulated by setting large adjustment values on the final expiry point." from 7.4.2.2.3

Remove "and provides the type of protection given by the OSEKtime Interrupt re-enable
schedule event [16]." from 7.7.2.1

Remove chapter 12.3

SWCT:
=====

Replace reference to OSEK COM:

Fifteen filter algorithms formally described by the enumeration type DataFilterTypeEnum in the meta-model are taken from OSEK COM 3.0.3 specification [18] that is referenced by the RTE specification [2].

by:

Fifteen filter algorithms formally described by the enumeration type DataFilterTypeEnum in the meta-model are taken from the ISO 17356-4 specification [18] that is referenced by the RTE specification [2].

Replace:

[TPS_SWCT_01222] Applicability of DataFilter
This OSEK specification states that filtering is only used for messages that can be interpreted as C language unsigned integer types (characters, unsigned integers and enumerations).(RS_SWCT_03221)

by:

[TPS_SWCT_01222] Applicability of DataFilter
The ISO 17356-4 specification states that filtering is only used for messages that can be interpreted as C language unsigned integer types (characters, unsigned integers and enumerations).(RS_SWCT_03221)

Replace:

[constr_1044] Applicability of DataFilter
According to the origin of DataFilter, i.e. OSEK COM 3.0.3 specification [18], DataFilters can only be applied to values with an integer base type.()

by:

[constr_1044] Applicability of DataFilter
According to the origin of DataFilter, i.e. ISO 17356-4 specification [18], DataFilters can only be applied to values with an integer base type.()

Replace footnote to [constr_1090]:

This constraint is valid at least in the OSEK standard where an extended task (that can have wait points) can only exist a single time in the context of the scheduler.

by:

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

============
Dem
============
Replace the reference [17] Communication in Chapter 3.1 Input documents & related standards and norms Bibliography by : ISO 17356-3 in www.iso.org/

============
Dcm
============
Replace the reference [8]Communication in Chapter 3.1 Input documents & related standards and norms Bibliography by : ISO 17356-3 in www.iso.org/

============
EXP_VFB

Add an entry in chapter "13 References" related to ISO 17356-4:

ISO 17356-4
OSEK/VDX Communication (COM)
www.iso.org

Change the references to "OSEK-COM V3.0.3" in Table 4.2 (4.3.2 From the point of view of the receiver) and EXP_Vfb_00028 (4.3.4 Filtering between the sender and the receiver) to "ISO 17356-4". Link the references to the entry in chapter "13 References".

============
SRS_BSWGeneral

5 General Requirements on Basic Software

Replace "OSEK OS" by "ISO 17356-3"

5.2.3.4 Standard header Files
Replace in [SRS_BSW_00348]

Because E_OK is already defined within OSEK OS, E_OK has to be checked for being already defined:
/* for OSEK compliance this typedef has been added */

by

Because E_OK is already defined within ISO 17356-3, E_OK has to be checked for being already defined
/* for ISO 17356 compliance this typedef has been added */

6.2 Related Standards and Norms

Remove
6.2.1 OSEK
[STD_OSEK_OS] OSEK/VDX Operating System Specification
http://www.osek-vdx.org

Replace with ISO 17356-3 norm

============
SRS_COM

1)

Remove [DOC_OSEK_GLOS] and all its references, since a) AUTOSAR has an own wording and glossary b) the references to (old) OSEK-terms most probably create more confusion than guidance

2)
Set reference of [DOC_OSEK_COM] to: ISO 17356-4: COM
Move refernce to Chapter 7.3 ISO and remove Chapter 7.2 OSEK

3)
Remove secton references to [DOC_OSEK_COM], just keep the textual references no numbers

4)
Search and replace textual references to OSEK COM 3.0.3 by ISO 17356-4: COM

5)
[SRS_Com_02084]: Change following text in Description,
from
< The possibilities to define those conditions shall be the same as defined in [DOC_OSEK_COM] reception filter algorithms (see [DOC_OSEK_GLOS], Section 2.2.2).
to
> The possibilities to define those conditions shall be the same as defined in [DOC_ISO_COM] reception filter algorithms (see [DOC_ISO_COM], Section 3.2.3).

6)
[SRS_Com_02058]: Change following text in Supporting Material,
from
< If no update bits are used, the AUTOSAR COM module provides the deadline monitoring defined in [DOC_OSEK_COM] (Section 2.5.1).
to
> If no update bits are used, the AUTOSAR COM module provides the deadline monitoring defined in [DOC_ISO_COM] (Section 3.5.1).

============
SRS_Os

in chapter 6.2.1 OSEK:

Replace "[STD_OSEK_OS] OSEK/VDX Operating System, Version 2.2.3, http://www.osek-vdx.org/mirror/os223.pdf"
by "[STD_OSEK_OS] ISO 17356-3: OS"

Replace "[STD_OSEK_OIL] OSEK / VDX Implementation Language (OIL) V2.5, OSEK Implementation Language, http://www.osek-vdx.org/mirror/oil25.pdf"
by "[STD_OSEK_OIL] ISO 17356-6: OIL"

Remove "[STD_OSEK_TTOS] OSEK/VDX Time-Triggered Operating System, Version 1.0, July 24, 2001, http://www.osek-vdx.org/mirror/ttos10.pdf"

Remove "[STD_OSEK_ORTI] OSEK/VDX ORTI (OSEK RunTime Interface) Part A Version 2.1.1, Part B Version 2.1, http://www.osek-vdx.org/mirror/ORTI-A-211.pdf"

in [SRS_Os_11002]: remove [STD_OSEK_TTOS] from Supporting Material

============
SWS_COM

1)
Set reference of [17] to ISO 17356-4: COM
Set reference of [18] to ISO 17356-6: OIL

2)
Search and replace textual references to OSEK COM 3.0.3 by ISO 17356-4: COM or [17] (if suitable)
and textual references to OSEK OIL by ISO 17356-6: OIL or [18] (if suitable)

============
SWS_StandardTypes:
In Section 3.2: replace
[7] OSEK/VDX Operating System, Version 2.2.2 www.osek-vdx.org/os222.pdf
by
[7] OSEK/VDX Operating System, ISO 17356-3: OS
============
CP_TR_AutosarModelConstraints
see PS for the SWCT.
============

==
SRS_NetworkManagement:

1) Change Section 7.2.1 name from "OSEK" to "ISO 17356-5"
Remove the description:
[5] [STD_OSEK_NM]

OSEK/VDX NM Specification (ISO 17356-5), Version 2.5.3
[STD_OSEK_NM] OSEK/VDX NM Specification (ISO 17356-5), V2.5.3
http://www.osek-vdx.org/
Change to :
OSEK/VDX NM Specification
www.iso.org
===>
7.2.1 ISO 17356-5
[5] ISO 17356-5: NM Specification
www.iso.org

2) Replace "OSEK-NM" to "ISO 17356-5: NM Specification" in [SRS_Nm_02515]

3) Replace "OSEK NM 2.5.3" to "ISO 17356-5: NM Specification" in [SRS_Nm_00142]
–Last change on issue 73564 comment 28–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.41 Specification Item TPS_SWCT_01225

Trace References:

none

Content:

Please note that it is technically possible to let a A single RunnableEntity can implement the functionality of two or more more than one ClientServerOperations. For this purposetwo or more , one OperationInvokedEvents need to reference this single Event for each affected ClientServerOperation shall reference the respective RunnableEntity.

In this case, however, it is essential that the queue length associated with each of the ClientServerOperations has the same value. In other words: The attribute ServerComSpec.ServerComSpec.queueLength shall be taken for the determination of the resulting queue length, constr_1128 applies.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77673: Handling of multiple ServerComSpec queueLength if SWS_Rte_08002 applies

Problem description:

Please clarify how to handle multiple ServerComSpec.queueLength in case [SWS_Rte_08002] applies, i.e. several operations are mapped to the same server runnable resulting in a single server invocation queue.

What is the resulting queue size?

In my opinion it would make sense to take the sum of all ServerComSpec.queueLengths, but SWCT [constr_1128] could be interpreted as to take only one of the queueLengths (which all have the same value anyway). Otherwise, I don't get the intention behind [constr_1128].

Btw: Should [SWS_Rte_08001] and [SWS_Rte_08002] better say "If two or more operations..." instead of "If two operations..."?

Agreed solution:

– TPS SWC-T –

Rephrase [TPS_SWCT_01225]:

[TPS_SWCT_01225] RunnableEntity implements the functionality of more than one ClientServerOperations
A single RunnableEntity can implement the functionality of more than one ClientServerOperations. For this purpose, one OperationInvokedEvent for each affected ClientServerOperation shall reference the respective RunnableEntity. The attribute ServerComSpec.queueLength shall be taken for the determination of the resulting queue length, constr_1128 applies.

– SWS RTE –

[SWS_Rte_08001] d If multiple operations are mapped to the same RunnableEntity, and [SWS_Rte_04522] requires a call serialization, then the operation invoked events
shall be mapped to same task and they shall have the same position in task. Otherwise
the RTE Generator shall reject configuration. c(SRS_Rte_00019, SRS_Rte_00033)

[SWS_Rte_08002] d If multiple operations are mapped to the same RunnableEntity, and [SWS_Rte_04522] requires a call serialization, then a single queue is implemented
for invocations coming from any of the operations. c(SRS_Rte_00019,

SRS_Rte_00033)
–Last change on issue 77673 comment 6–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.42 Specification Item TPS_SWCT_01288

Trace References:

none

Content:

DataConstr is an ARElement which can be reused by several data type specifications. Especially an ImplementationDataType and an ApplicationDataType which are mapped to each other, can refer to the same constraints or they can define their own constraints.

To avoid conflicts, in both cases PhysConstrs shall be interpreted by tools only with respect to application data types while InternalConstrs shall be interpreted only with respect to implementation data types. If either a physical or internal constraint is missing an existing

If only PhysConstrs are provided to ApplicationDataTypes the CompuMethod can be used to calculate the missing informationcompute the InternalConstrs.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #74212: Clarifications on DataConstrs w.r.t. optionallity and PhysConstrs versus InternalConstr

  Problem description:

  The figure Figure 5.6: Value ranges and invalid values shows very nice how the different data constraints needs to match to each other.

  1.) But the figure shows something like intern constrains of an ApplicationDataType. How des this match to TPS_SWCT_01288 which implies that for ApplicationDataTypes only PhysConstrs are applicable. Or is this a misinterpretation?

  2.) Since DataConstr are optional, I miss also some precise statement how the limits of a data type is defined in case of missing DataConstr.

I would assume the following:
If an ApplicationDataTypes does not define a DataConstr the limit of the later mapped ImplementationDataType applies.
If an ImplementationDataType does not define a DataConstr the technical limit of the BaseType defined by the size of type in bits and the according encoding applies.

Agreed solution:

1. Split the dataConstr line in tables 5.8, 5.18, 5.32, and 5.33 such that two new line are created

dataConstr.physConstr
dataConstr.internalConstr

Keep the existing setting for the respective flavor (application vs. implementation) and set the respective other line to to "d/c" and add an explanation to the legend that "d/c" stands for "don't care.

Update TPS_SWCT_01288:

[TPS_SWCT_01288] Interpretation of PhysConstrs and InternalConstrs by tools
DataConstr is an ARElement which can be reused by several data type specifications. Especially an ImplementationDataType and an Application-
DataType which are mapped to each other, can refer to the same constraints or they can define their own constraints.
To avoid conflicts, in both cases PhysConstrs shall be interpreted by tools only with respect to application data types while InternalConstrs shall be interpreted only with respect to implementation data types.
In case only physConstr are provided to ApplicationDataTypes the CompuMethod can be used to compute the internalConstr. ()

In Figure 5.6 (Value ranges and invalid values), replace the string "internalConstrs of ApplicationDataType" by "computed internalConstrs of ApplicationDataType"

2. Add note: If an ApplicationPrimitiveDataType does not define dataConstr then implicit constraints can be derived from physical meaning of the ApplicationDataType. For example, if the data type represents a temperature, the lower bound will not be able to exceed 0K. For other physical meanings it could be possible that the implicitly assumed limits go from -INF to +INF. In order to void ambiguity for the values of limits it is strongly recommended to define a reasonable limit for ApplicationPrimitiveDataTypes.
–Last change on issue 74212 comment 7–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.43 Specification Item TPS_SWCT_01444

Trace References:

none

Content:

In both cases (mentioned in constr_1012and constr_1013) the size of SwBaseType is specified in bits.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #74446: Explanation for the applicability of VARIABLE_LENGTH missing

  Problem description:

  The SWCT contains constr_1011 and constr_1013 that mention the value VARIABLE_LENGTH as a possible setting for SwBaseType.category.

  There is no clear description when this value would be applicable. In fact, doubts are warranted whether the value (and with it, the attribute maxBaseTypeSize) should be applicable in AUTOSAR at all.

  Background: the definition of a SwBaseType is supposed to refer to an atomic data type. The "base type" for an array is still the "base type" of the array elements, as opposed to the definition of a base type for the array itself.

  The difference between the definition of a "base type" for the array elements and a "base type" for the entire array manifests in the attributes baseTypeSize, resp. maxBaseTypeSize.

  In some cases definitions of SwBaseType can be found in the wild that look similar to this specimen:

  ```
  <SW-BASE-TYPE>
  <SHORT-NAME>FunnyBaseType</SHORT-NAME>
  <CATEGORY>VARIABLE_LENGTH</CATEGORY>
  <MAX-BASE-TYPE-SIZE>1433</MAX-BASE-TYPE-SIZE>
  <BASE-TYPE-ENCODING>WINDOWS-1252</BASE-TYPE-ENCODING>
  </SW-BASE-TYPE>
  ```

I guess that this base type has been build on the assumption that a string deserves a base type for the entire Windows-1252-encoded string.

This assumption would not only result in countless definitions of SwBaseTypes for each possible combination of array length and data type, it would also create an unholy dependency to the value of ImplementationDataTypeElement.arraySize. In other words, if the value of array size changes than the value of maxBaseTypeSize would have to change accordingly.

This does not make much sense.

I tried to get some information about the use case for VARIABLE_LENGTH that was originally intended. Bernhard mentioned that it was meant to support edge cases for the generation of A2L files, in particular for string implementations where the first element denotes the length and the rest is reserved for the payload.

If there are further known use cases, please feel free to leave a comment ...

I would like to provide a better explanation to the applicability of VARIABLE_LENGTH and also better explain that the SwBaseType is not meant to be applied to composite data types.

At the moment the document is rather indifferent about these two aspects.

Agreed solution:

SWCT:
—–

1. Reformulate:

[constr_1011] category of SwBaseType
d The attribute SwBaseType.category must be set and only the values FIXED_LENGTH and VOID are supported. c

[constr_1422] Value of category is VOID
d If the value of the attribute SwBaseType.category is set to VOID then the attribute baseTypeSize shall not exist. c()

[constr_1012] Value of category is FIXED_LENGTH
d If the value of the attribute SwBaseType.category is set to FIXED_LENGTH then the attribute baseTypeSize shall be filled with content. c()

[constr_1229] category of ImplementationDataType boils down to VALUE
subElement.category is set to VALUE or TYPE_REFERENCE that eventually boils down to VALUE and the subElement refers to a SwBaseType where baseTypeSize is set to the value 8 and the baseTypeEncoding is set to NONE.

[constr_1220] Compatibility of SwBaseType
d Two SwBaseTypes are compatible if and only if attributes baseTypeSize respectively byteOrder, memAlignment, baseTypeEncoding, and nativeDeclaration have identical values. c()

2. Remove [constr_1013].

3. Set attribute maxBaseTypeSize to ’obsolete’.

4. Adapt Listing 5.4: Example for the definition of a string SwBaseType: (add CATEGORY)

```
<AR-PACKAGE>
<ELEMENTS>
<SW-BASE-TYPE>
<SHORT-NAME>MyTextBaseType</SHORT-NAME>
<CATEGORY>FIXED_LENGTH</CATEGORY>
<BASE-TYPE-SIZE>8</BASE-TYPE-SIZE>
<BASE-TYPE-ENCODING>UTF-8</BASE-TYPE-ENCODING>
</SW-BASE-TYPE>
<SW-BASE-TYPE>
<SHORT-NAME>uint8BaseType</SHORT-NAME>
<CATEGORY>FIXED_LENGTH</CATEGORY>
<BASE-TYPE-SIZE>8</BASE-TYPE-SIZE>
</SW-BASE-TYPE>
</ELEMENTS>
</AR-PACKAGE>
```

5. Fix all object diagrams showing SwBaseType (add CATEGORY).

DEXT:
—–

1. Remove in Table 4.8: Allowed attributes of SwDataDefProps for DiagnosticDataElement.swDataDefProps the line baseType.baseTypeDefinition.maxBaseTypeSize N/A.
–Last change on issue 74446 comment 7–

BW-C-Level:

Application	Specification	Bus
3	3	1

1.44 Specification Item TPS_SWCT_01490

Trace References:

none

Content:

As of now, AUTOSAR supports (as depicted by Figure REF fig_3a_AUTOSAR_Meta_2d_Model_DOC_ErrorsAndInterfaces) ApplicationErrors only for ClientServerInterfaces.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76526: TPS_SWCT_01490 is no longer true, thanks to the adaptive platform

  Problem description:

  Because ApplicationErrors are also used on the adaptive platform, TPS_SWCT_01490 and constr_1038 are no longer true in its literal meaning.

  Either we remove the specification item or else we should add some further qualification such that the statement is explicitly limited to the classic platform.
  –Last change on issue 76526 comment 3–

  Agreed solution:

  Remove TPS_SWCT_01490

  Update the existing phrasing of const_1038

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

  to

  [constr_1038] Reference to ApplicationError
  A possibleError referenced by a ClientServerOperation shall be owned by the

PortInterface that also owns the ClientServerOperation.
–Last change on issue 76526 comment 4–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.45 Specification Item TPS_SWCT_01560

Trace References:

RS_SWCT_03210

Content:

Data conversion shall be supported for AutosarDataTypes that refer to CompuMethods of Identifiable.category LINEAR, IDENTICAL,

- LINEAR,
- IDENTICAL,
- SCALE_LINEAR_AND_TEXTTABLE, TEXTTABLE, and
- TEXTTABLE,
- BITFIELD_TEXTTABLE. , and
- RAT_FUNC - as long as the semantics of the latter comes down to a reciprocal linear data scaling.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #74570: Contradiction w.r.t data conversion [TPS_SWCT_01560] & [SWS_Rte_07928]

  Problem description:

  There seems to be a discrepency between SWC-T and RTE w.r.t data conversion. Indeed in the TPS_SWCT following is stated

  [TPS_SWCT_01560] Supported categorys of CompuMethods for data conversion d Data conversion shall be supported for AutosarDataTypes that refer to CompuMethods of category LINEAR, IDENTICAL, SCALE_LINEAR_AND_TEXTTABLE, TEXTTABLE, and BITFIELD_TEXTTABLE. c(RS_SWCT_03210)

  whereas RTE allows the data conversion for compuMethods of category

RAT_FUNC.

[SWS_Rte_07928] d The data conversion shall be supported for data
types that refer to CompuMethods of category LINEAR, IDENTICAL,
SCALE_LINEAR_AND_TEXTTABLE, TEXTTABLE , BITFIELD_TEXTTABLE and
CompuMethods of category RAT_FUNC with a reciprocal linear data scaling.
c(SRS_Rte_00182)

it shall be clearly specified if data conversion for RAT_FUNC (i.e. LINEAR/RAT_FUNC, RAT_FUNC/LINEAR, RAT_FUNC/RAT_FUNC,)is supported or not.

Agreed solution:

Replace

[TPS_SWCT_01560] Supported categorys of CompuMethods for data conversion
Data conversion shall be supported for AutosarDataTypes that refer to CompuMethods of category LINEAR, IDENTICAL, SCALE_LINEAR_AND_TEXTTABLE, TEXTTABLE, and BITFIELD_TEXTTABLE.

by

[TPS_SWCT_01560] Supported categorys of CompuMethods for data conversion
Data conversion shall be supported for AutosarDataTypes that refer to CompuMethods of category LINEAR, IDENTICAL, SCALE_LINEAR_AND_TEXTTABLE, TEXTTABLE, BITFIELD_TEXTTABLE, and RAT_FUNC, as long as semantics of the latter comes down to a reciprocal linear data scaling.
–Last change on issue 74570 comment 6–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.46 Specification Item TPS_SWCT_01675

Trace References:

RS_SWCT_03225

Content:

The formal modeling of a NvBlockDescriptor should follow the recommendations given in table [REF table_3a_NvBlockNeeds_20_dependencies]. 7204072420 720407242076548

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #74703: ramBlockStatusControl recommendation in table 7.11

  Problem description:

  Table 7.11

  ramBlockStatusControl
  Can be different.
  Recommended to set to true if any of the nv data PortPrototypes requests a use of the API for accessing the block.

  ramBlockStatusControl is an enum so i believe that the recommendation is wrong

  Agreed solution:

  In table 11.7 ("NvBlockNeeds dependencies"), replace the text in the third column of the row starting with "ramBlockStatusControl" by

  Recommended to set to the value RamBlockStatusControlEnum.api if any of the nv data PortPrototypes requests a use of the API for accessing the block.
  –Last change on issue 74703 comment 3–

  BW-C-Level:

Application	Specification	Bus
1	1	1

1.47 Specification Item TPS_SWCT_01678

Trace References:

none

Content:

ServiceNeeds Kind : SyncTimeBaseMgrUserNeeds

RoleBasedPortAssignment valid roles:

- StbM_TimeBaseValue StartTimer [1]
- TimeoutNotification [0..1]

RoleBasedDataAssignment N/A

RoleBasedDataTypeAssignment N/A

RepresentedPortGroups N/A

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76867: Role values in TPS_SWCT_01678 and TPS_SWCT_01679

  Problem description:

  TPS_SWCT_01678 and TPS_SWCT_01679 provide service use cases for the interaction of software-components with the StbM.

  The values of the role attribute of a RoleBasedPortAssignment are supposed to indicate the PortInterface used for the related PortPrototype.

  The problem is that none of the values of the role attribute correspond to existing PortInterfaces in the SWS StbM.

  In particular, the document assumes the existence of three ClientServerInterfaces:

  * StbM_TimeBaseValue
  * StbM_TimeBase_TriggerCustomer
  * StbM_TimeBase_StateNotification

  The same PortInterfaces are also mentioned in the EXP_VFB document (which consequently is added to the impact).

  The StbM actually provides these ClientServerInterfaces:

  * StbM_GlobalTime_Master
  * StbM_GlobalTime_Slave
  * StbM_StartTimer
  * StbM_TimeNotification
  * StbM_MesurementNotification

  Also, one SenderReceiverInterface is available:

* StbM_StatusNotification

This more or less means that the service use cases need to be redefined such that the usage of the existing service interfaces is described.

Agreed solution:

VFB
===

Update R17.14 (11.2.1 Intra-ECU features):
A ComponentPrototype is requiring some StbM Interfaces (example: ClientServerInterface StartTimer or SenderReceiverInterface StatusNotification).

SWCT
====

Add note to chapter 13.10, above chapter 13.10.1:

The general idea behind the time synchronization concept is that the role of global time master and global time slave are partly implemented in the application software. For this purpose, the application software provides PortPrototypes typed by the standardized PortInterfaces GlobalTime_Master and GlobalTime_Slave.

In many cases both PortInterfaces will be used by the application software of one ECU. This means that the ECU is a global time slave on one domain and a global time master on another domain.

In terms of modeling, a given global time domain is represented by a SwcServiceDependency. If one software-component has to deal with different global time domains (e.g. because it represents a slave in one domain and a master in another) then the corresponding SwcInternalBehavior needs to define one SwcServiceDependency per global time domain.

Rework chapter 13.10.1:

StbM use case: start timer and potentially get notified about its expiration

Scenario: a software-component wants to wind up a timer in the StbM with a given expiration time. The software-component may want to receive a notification when the timer expires. In this case the following setup applies:

[TPS_SWCT_01678] StbM use case: start timer and potentially get notified about its expiration

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
StartTimer [1]
TimeNotification [0..1]
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

In this case the software-component needs to have an RPortPrototype typed by the ClientServerInterface StartTimer and (if applicable) a PPortPrototype typed by the ClientServerInterface TimeNotification.

Rework chapter 13.10.2:

StbM Use Case: Software-Components wants to get notifications of status changes

Scenario: a software-component wants to receive events whenever the status of the StbM changes. For this purpose, the software-component sports a sender/receiver RPortPrototype.

[TPS_SWCT_01679] StbM Use Case: Software-Components wants to get notifications of status changes

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
StatusNotification [1]
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A

()

The events received from the StbM have a fixed structure. For more details, please refer to [SWS_StbM_00284]

New chapter: StbM Use Case: Process time snapshot obtained from global time slave for diagnostics purposes

Scenario: a software-component provides a PPortPrototype onto which the global time slave pushes snapshots of time synchronization records. This data is typically used for diagnostic purposes.

[TPS_SWCT_0xxx1] StbM Use Case: Process time snapshot obtained from global time slave for diagnostics purposes

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
MeasurementNotification [1]
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

Note that in this case the SWC acts as a server, the StbM implements a client role!

[TPS_SWCT_0xxx2] Suffix used for the resulting name of the PortInterface for measurement notification
The suffix used for the resulting name of the PortInterface for the measurement notification (MeasurementNotification_TB_Name) shall be taken from the shortName of the applicable SwcServiceDependency.

New chapter: StbM Use Case: software-component represents a global time master

Scenario: a software-component implements the application-software part of the global time master role. For this purpose the software-component exposes an RPortPrototype that is supposed to be connected to the StbM service component.

[TPS_SWCT_0xxx3] StbM Use Case: software-component represents a global time master

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
GlobalTime_Master [1]
StatusNotification [0..1]
MeasurementNotification [0..1
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

[TPS_SWCT_0xxx4] Suffix used for the resulting name of the PortInterface for the global time master role
The suffix used for the resulting name of the PortInterface for the global time master role (GlobalTime_Master_Name) shall be taken from the shortName of the applicable SwcServiceDependency.

New chapter: StbM Use Case: software-component represents a global time slave

Scenario: a software-component implements the application-software part of the global time slave role. For this purpose the software-component exposes an RPortPrototype that is supposed to be connected to the StbM service component.

[TPS_SWCT_0xxx5] StbM Use Case: software-component represents a global time slave

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
GlobalTime_Slave [1]
StatusNotification [0..1]
MeasurementNotification [0..1]
RoleBasedDataAssignment
N/A

RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

[TPS_SWCT_0xxx6] Suffix used for the resulting name of the PortInterface for the global time slave role
The suffix used for the resulting name of the PortInterface for the global time slave role (GlobalTime_Slave_Name) shall be taken from the shortName of the applicable SwcServiceDependency.
–Last change on issue 76867 comment 12–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.48 Specification Item TPS_SWCT_01679

Trace References:

none

Content:

ServiceNeeds Kind : SyncTimeBaseMgrUserNeeds

RoleBasedPortAssignment valid roles:

- StbM_TimeBase_TriggerCustomer 0..1
- StbM_TimeBase_StateStatusNotification [0..11]

RoleBasedDataAssignment N/A

RoleBasedDataTypeAssignment N/A

RepresentedPortGroups N/A

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76867: Role values in TPS_SWCT_01678 and TPS_SWCT_01679

  Problem description:

  TPS_SWCT_01678 and TPS_SWCT_01679 provide service use cases for the interaction of software-components with the StbM.

The values of the role attribute of a RoleBasedPortAssignment are supposed to indicate the PortInterface used for the related PortPrototype.

The problem is that none of the values of the role attribute correspond to existing PortInterfaces in the SWS StbM.

In particular, the document assumes the existence of three ClientServerInterfaces:

* StbM_TimeBaseValue
* StbM_TimeBase_TriggerCustomer
* StbM_TimeBase_StateNotification

The same PortInterfaces are also mentioned in the EXP_VFB document (which consequently is added to the impact).

The StbM actually provides these ClientServerInterfaces:

* StbM_GlobalTime_Master
* StbM_GlobalTime_Slave
* StbM_StartTimer
* StbM_TimeNotification
* StbM_MesurementNotification

Also, one SenderReceiverInterface is available:

* StbM_StatusNotification

This more or less means that the service use cases need to be redefined such that the usage of the existing service interfaces is described.

Agreed solution:

VFB
===

Update R17.14 (11.2.1 Intra-ECU features):
A ComponentPrototype is requiring some StbM Interfaces (example: ClientServerInterface StartTimer or SenderReceiverInterface StatusNotification).

SWCT
====

Add note to chapter 13.10, above chapter 13.10.1:

The general idea behind the time synchronization concept is that the role of global time master and global time slave are partly implemented in the application software. For this purpose, the application software provides PortPrototypes typed by the standardized PortInterfaces GlobalTime_Master and GlobalTime_Slave.

In many cases both PortInterfaces will be used by the application software of one ECU. This means that the ECU is a global time slave on one domain and a global time master on another domain.

In terms of modeling, a given global time domain is represented by a SwcServiceDependency. If one software-component has to deal with different global time domains (e.g. because it represents a slave in one domain and a master in another) then the corresponding SwcInternalBehavior needs to define one SwcServiceDependency per global time domain.

Rework chapter 13.10.1:

StbM use case: start timer and potentially get notified about its expiration

Scenario: a software-component wants to wind up a timer in the StbM with a given expiration time. The software-component may want to receive a notification when the timer expires. In this case the following setup applies:

[TPS_SWCT_01678] StbM use case: start timer and potentially get notified about its expiration

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
StartTimer [1]
TimeNotification [0..1]
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

In this case the software-component needs to have an RPortPrototype typed by the ClientServerInterface StartTimer and (if applicable) a PPortPrototype typed by the ClientServerInterface TimeNotification.

Rework chapter 13.10.2:

StbM Use Case: Software-Components wants to get notifications of status changes

Scenario: a software-component wants to receive events whenever the status of the StbM changes. For this purpose, the software-component sports a sender/receiver RPortPrototype.

[TPS_SWCT_01679] StbM Use Case: Software-Components wants to get notifications of status changes

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
StatusNotification [1]
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

The events received from the StbM have a fixed structure. For more details, please refer to [SWS_StbM_00284]

New chapter: StbM Use Case: Process time snapshot obtained from global time slave for diagnostics purposes

Scenario: a software-component provides a PPortPrototype onto which the global time slave pushes snapshots of time synchronization records. This data is typically used for diagnostic purposes.

[TPS_SWCT_0xxx1] StbM Use Case: Process time snapshot obtained from global time slave for diagnostics purposes

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
MeasurementNotification [1]
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

Note that in this case the SWC acts as a server, the StbM implements a client role!

[TPS_SWCT_0xxx2] Suffix used for the resulting name of the PortInterface for measurement notification
The suffix used for the resulting name of the PortInterface for the measurement notification (MeasurementNotification_TB_Name) shall be taken from the shortName of the applicable SwcServiceDependency.

New chapter: StbM Use Case: software-component represents a global time master

Scenario: a software-component implements the application-software part of the global time master role. For this purpose the software-component exposes an RPortPrototype that is supposed to be connected to the StbM service component.

[TPS_SWCT_0xxx3] StbM Use Case: software-component represents a global time master

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
GlobalTime_Master [1]
StatusNotification [0..1]
MeasurementNotification [0..1
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

[TPS_SWCT_0xxx4] Suffix used for the resulting name of the PortInterface for the global time master role
The suffix used for the resulting name of the PortInterface for the global time master role (GlobalTime_Master_Name) shall be taken from the shortName of the applicable SwcServiceDependency.

New chapter: StbM Use Case: software-component represents a global time slave

Scenario: a software-component implements the application-software part of the global time slave role. For this purpose the software-component exposes an RPortPrototype that is supposed to be connected to the StbM service component.

[TPS_SWCT_0xxx5] StbM Use Case: software-component represents a global time slave

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
GlobalTime_Slave [1]
StatusNotification [0..1]
MeasurementNotification [0..1]
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

[TPS_SWCT_0xxx6] Suffix used for the resulting name of the PortInterface for the global time slave role
The suffix used for the resulting name of the PortInterface for the global time slave role (GlobalTime_Slave_Name) shall be taken from the shortName of the applicable SwcServiceDependency.
–Last change on issue 76867 comment 12–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.49 Specification Item TPS_SWCT_01714

Trace References:

RS_SWCT_00120, RS_SWCT_02090

Content:

If the value of attribute SwcExclusiveAreaPolicy.SwcExclusiveAreaPolicy.apiPrinciple is set to ApiPrincipleEnum.ApiPrincipleEnum.common perExecutable then the RTE provides individual sets of APIs for entering and exiting ExclusiveAreas for each affected Runnable Entity.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76898: Wrong value of ApiPrincipleEnum in TPS_SWCT_01714

  Problem description:

  Name: Ulrich Kiffmeier
  Phone: +49 5251 1638 747
  Role: WP-M member

  Description/Motivation:

  TPS_SWCT_01714 reads:

  [TPS_SWCT_01714] ExclusiveArea is entered and exited by an individual set of APIs
  If the value of attribute SwcExclusiveAreaPolicy.apiPrinciple is set to ApiPrincipleEnum.common then the RTE provides individual sets of APIs for entering and exiting ExclusiveAreas for each affected RunnableEntity.
  (RS_SWCT_00120, RS_SWCT_02090)

  I think the spec item is about ApiPrincipleEnum.perExecutable not ApiPrincipleEnum.common. Seems to be a copy& paste error from TPS_SWCT_01713.

  Agreed solution:

  Replace the current formulation of TPS_SWCT_01714 with this:

  [TPS_SWCT_01714] ExclusiveArea is entered and exited by an individual set of APIs
  If the value of attribute SwcExclusiveAreaPolicy.apiPrinciple is set to ApiPrincipleEnum.perExecutable then the RTE provides individual sets of APIs for entering

and exiting ExclusiveAreas for each affected RunnableEntity. (RS_SWCT_00120, RS_SWCT_02090)
–Last change on issue 76898 comment 3–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.50 Specification Item TPS_SWCT_01736

Trace References:

RS_SWCT_02100

Content:

If a Unit does not define the attribute Unit.Unit.physicalDimension the default Physical Dimension with the Referrable.shortNameNoDimension applies where all physical exponents are set to 0.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #74210: Clarify handling of PhysicalDimensions

  Problem description:

  1.)TPS_SWCT_01281 defines:
  ****
  [TPS_SWCT_01281] Unit associated with a PhysicalDimension d A unit
  (described in next section) can be associated with a PhysicalDimension. c()
  ****

  But what happens if an Unit is NOT associated with a PhysicalDimension?

  I would assume that AUTOSAR defines in this case the assumed default Physical Dimension E.g. the NoDimension from Application Interface
  <PHYSICAL-DIMENSION>
  <SHORT-NAME>NoDimension</SHORT-NAME>
  <LONG-NAME>
  <L-4 L="EN">No Dimension</L-4>
  </LONG-NAME>
  </PHYSICAL-DIMENSION>

  2) all SI exponents of an PhysicalDimension are optional and also the Appli-

cation Interface Physical dimension do only describe SI exponents different from 0
I miss the explicit statement that the default value of SI exponents are 0

3.) By the way in the statement [TPS_SWCT_01549]
***
if the PhysicalDimension is only defined on one side (sender or receiver)
then it shall be considered as default for the other side.
***
the meaning of "as default" could be clarified / linked to the clarification of 1.) and 2.)
–Last change on issue 74210 comment 1–

Agreed solution:

[TPS_SWCT_0xxx1] Default values for Unit.physicalDimension
If a Unit does not define the attribute Unit.physicalDimension the default PhysicalDimension with the shortName NoDimension applies where all physical exponents are set to 0.

[TPS_SWCT_0xxx2] Default values for physical exponents

The default value of attributes currentExp, lengthExp, luminousIntensityExp, massExp, molarAmountExp, temperatureExp, timeExp is 0.
–Last change on issue 74210 comment 6–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.51 Specification Item TPS_SWCT_01737

Trace References:

RS_SWCT_02100

Content:

The default value of attributes PhysicalDimension.currentExp, PhysicalDimension.lengthExp, PhysicalDimension.luminousIntensityExp, PhysicalDimension.massExp, PhysicalDimension.molarAmountExp, PhysicalDimension.temperatureExp, PhysicalDimension.timeExp is 0.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #74210: Clarify handling of PhysicalDimensions

Problem description:

1.)TPS_SWCT_01281 defines:
****
[TPS_SWCT_01281] Unit associated with a PhysicalDimension d A unit (described in next section) can be associated with a PhysicalDimension. c()
****

But what happens if an Unit is NOT associated with a PhysicalDimension?

I would assume that AUTOSAR defines in this case the assumed default Physical Dimension E.g. the NoDimension from Application Interface
<PHYSICAL-DIMENSION>
<SHORT-NAME>NoDimension</SHORT-NAME>
<LONG-NAME>
<L-4 L="EN">No Dimension</L-4>
</LONG-NAME>
</PHYSICAL-DIMENSION>

2) all SI exponents of an PhysicalDimension are optional and also the Application Interface Physical dimension do only describe SI exponents different from 0
I miss the explicit statement that the default value of SI exponents are 0

3.) By the way in the statement [TPS_SWCT_01549]
***
if the PhysicalDimension is only defined on one side (sender or receiver)
then it shall be considered as default for the other side.
***
the meaning of "as default" could be clarified / linked to the clarification of 1.) and 2.)
–Last change on issue 74210 comment 1–

Agreed solution:

[TPS_SWCT_0xxx1] Default values for Unit.physicalDimension
If a Unit does not define the attribute Unit.physicalDimension the default PhysicalDimension with the shortName NoDimension applies where all physical exponents are set to 0.

[TPS_SWCT_0xxx2] Default values for physical exponents

The default value of attributes currentExp, lengthExp, luminousIntensityExp, massExp, molarAmountExp, temperatureExp, timeExp is 0.
–Last change on issue 74210 comment 6–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.52 Specification Item TPS_SWCT_01738

Trace References:

none

Content:

The references in the roles

- ArParameterInImplementationDataInstanceRef.portPrototype
- ArParameterInImplementationDataInstanceRef.rootParameterDataPrototype
- ordered collection of ArParameterInImplementationDataInstanceRef.contextDataPrototype
- ArParameterInImplementationDataInstanceRef.targetDataPrototype

constitute the path leading from the root to the specified inner instance of a ParameterInterface.parameter inside of a ParameterDataPrototype typed by an ImplementationDataType.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76467: SubElementMapping for ParameterDataProtypes typed by ImplementationDataType

  Problem description:

  VariableAndParameterInterfaceMapping breeds DataPrototypeMapping.
  DataPrototypeMapping breeds SubElementMapping.
  SubElementMapping breeds ImplementationDataTypeSubElementRef.

  And ImplementationDataTypeSubElementRef is the path to the dark side.

  Because ...

  ImplementationDataTypeSubElementRef refers to VariableDataPrototype in the role rootVariableDataPrototype.

  ImplementationDataTypeSubElementRef does NOT refer to ParameterDataPrototype in an applicable role to be used as the root element.

This rules out the application of the ImplementationDataTypeSubElementRef for a ParameterDataPrototype that takes the role of a root element.

And again, what we actually wanted to implement with this mess is called VariableAnd**Parameter**InterfaceMapping.

Sigh.

Agreed solution:

Add new meta-class named ArParameterInImplementationDataInstanceRef (decription: <come up something creative out of the description of the ArVariableInImplementationDataInstanceRef>)

Aggregate ArParameterInImplementationDataInstanceRef at ImplementationDataTypeSubElementRef with the mulitplicity 0..1 in the role parameterImplementationDataTypeElement.

Change the multiplicity of the aggregation ImplementationDataTypeSubElementRef.implementationDataTypeElement to 0..1

Add reference from ArParameterInImplementationDataInstanceRef to ParameterDatapPototype in the role rootParameterDataPrototype. Multiplicity: 0..1

Add reference from ArParameterInImplementationDataInstanceRef to ImplementationDataTypeElement in the role contextDataPrototype. Multiplicity: 0..*

Add reference from ArParameterInImplementationDataInstanceRef to ImplementationDataTypeElement in the role targetDataPrototype. Mulitplicity: 1

add constraint:

constr_xxxx Existence of ImplementationDataTypeSubElementRef.implementationDataTypeElement vs. ImplementationDataTypeSubElementRef.parameterImplementationDataTypeElement
For any given ImplementationDataTypeSubElementRef, either the aggregation ImplementationDataTypeSubElementRef.implementationDataTypeElement or ImplementationDataTypeSubElementRef.parameterImplementationDataTypeElement shall exist.
–Last change on issue 76467 comment 5–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.53 Specification Item TPS_SWCT_01739

Trace References:

none

Content:

Scenario: an AtomicSwComponentType wants to react on suppressed or unavailable events and disable the permission to run for a FID. In this case, the following setup applies:

ServiceNeeds kind FunctionInhibitionAvailabilityNeeds

RoleBasedPortAssignment valid roles:

- ControlFunctionAvailable [1]

RoleBasedDataAssignment N/A

RepresentedPortGroups N/A

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #69989: add missing use case description for FiM EventAvailability

  Problem description:

  In RfC
  [RfC # 65847] [Fim] consider EventAvailability/EventSupression

  The interface ControlFunctionAvailable was added.

  But the service use case was not clarified and added to the SWC-T and BSWMD-T.

  Agreed solution:

  SWCT
  ====

  Add new subclass of ServiceNeeds: FunctionInhibitionAvailabilityNeeds, description: "Specifies the abstract needs on the configuration of the Function Inhibition Manager to provide the control function for one Function Identifier (FID)."

Add a reference from FunctionInhibitionAvailabilityNeeds to FunctionInhibitionNeeds in the role controlledFid, mulitplicity 0..1, description: "This reference represents the controlled FID".

Add chapter 13.8.2.2 Function Inhibition Manager Use Case: react on suppressed or unavailable events

[TPS_SWCT_0xxxx] Setup for Function Inhibition Manager Service use Case: react on suppressed or unavailable events
Scenario: an AtomicSwComponentType wants to react on suppressed or unavailable events and disable the permission to run for a FID.
In this case, the following setup applies:
ServiceNeedsKind: FunctionInhibitionAvailabilityNeeds
RoleBasedPortAssignment valid roles:
* ControlFunctionAvailable [1]
RoleBasedDataAssignment
N/A
RepresentedPortGroups
N/A

Note: for variant coding ClientServerInterface ControlFunctionAvailable is used to deactivate a certain functionality (e.g. to set the FID to not available)

For more information, please refer to [SWS_Fim_00107].

BSWMDT
======

Add chapter 13.2.2.1.2 Function Inhibition Manager Use Case: react on suppressed or unavailable events

[TPS_BSWMDT_0xxxx] Setup for Function Inhibition Manager Service use Case: react on suppressed or unavailable events
Scenario: a Basic Software Module wants to react on suppressed or unavailable events and disable the permission to run for a FID.
In this case, the following setup applies:
ServiceNeedsKind: FunctionInhibitionAvailabilityNeeds
RoleBasedPortAssignment valid roles:
* Fim_SetFunctionAvailable [1]
RoleBasedDataAssignment
N/A
RepresentedPortGroups

N/A

Note: for variant coding ClientServerInterface ControlFunctionAvailable is used to deactivate a certain functionality (e.g. to set the FID to not available)

For more information, please refer to [SWS_Fim_00106].
–Last change on issue 69989 comment 26–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.54 Specification Item TPS_SWCT_01740

Trace References:

none

Content:

ServiceNeeds Kind : SyncTimeBaseMgrUserNeeds

RoleBasedPortAssignment valid roles:

- MeasurementNotification [1]

RoleBasedDataAssignment N/A

RoleBasedDataTypeAssignment N/A

RepresentedPortGroups N/A

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76867: Role values in TPS_SWCT_01678 and TPS_SWCT_01679

  Problem description:

  TPS_SWCT_01678 and TPS_SWCT_01679 provide service use cases for the interaction of software-components with the StbM.

  The values of the role attribute of a RoleBasedPortAssignment are supposed to indicate the PortInterface used for the related PortPrototype.

  The problem is that none of the values of the role attribute correspond to existing PortInterfaces in the SWS StbM.

In particular, the document assumes the existence of three ClientServerInterfaces:

* StbM_TimeBaseValue
* StbM_TimeBase_TriggerCustomer
* StbM_TimeBase_StateNotification

The same PortInterfaces are also mentioned in the EXP_VFB document (which consequently is added to the impact).

The StbM actually provides these ClientServerInterfaces:

* StbM_GlobalTime_Master
* StbM_GlobalTime_Slave
* StbM_StartTimer
* StbM_TimeNotification
* StbM_MesurementNotification

Also, one SenderReceiverInterface is available:

* StbM_StatusNotification

This more or less means that the service use cases need to be redefined such that the usage of the existing service interfaces is described.

Agreed solution:

VFB
===

Update R17.14 (11.2.1 Intra-ECU features):
A ComponentPrototype is requiring some StbM Interfaces (example: ClientServerInterface StartTimer or SenderReceiverInterface StatusNotification).

SWCT
====

Add note to chapter 13.10, above chapter 13.10.1:

The general idea behind the time synchronization concept is that the role of global time master and global time slave are partly implemented in the application software. For this purpose, the application software provides PortPrototypes typed by the standardized PortInterfaces GlobalTime_Master and GlobalTime_Slave.

In many cases both PortInterfaces will be used by the application software of one ECU. This means that the ECU is a global time slave on one domain and a global time master on another domain.

In terms of modeling, a given global time domain is represented by a SwcServiceDependency. If one software-component has to deal with different global time domains (e.g. because it represents a slave in one domain and a master in another) then the corresponding SwcInternalBehavior needs to define one SwcServiceDependency per global time domain.

Rework chapter 13.10.1:

StbM use case: start timer and potentially get notified about its expiration

Scenario: a software-component wants to wind up a timer in the StbM with a given expiration time. The software-component may want to receive a notification when the timer expires. In this case the following setup applies:

[TPS_SWCT_01678] StbM use case: start timer and potentially get notified about its expiration

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
StartTimer [1]
TimeNotification [0..1]
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

In this case the software-component needs to have an RPortPrototype typed by the ClientServerInterface StartTimer and (if applicable) a PPortPrototype typed by the ClientServerInterface TimeNotification.

Rework chapter 13.10.2:

StbM Use Case: Software-Components wants to get notifications of status changes

Scenario: a software-component wants to receive events whenever the status of the StbM changes. For this purpose, the software-component sports a sender/receiver RPortPrototype.

[TPS_SWCT_01679] StbM Use Case: Software-Components wants to get notifications of status changes

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
StatusNotification [1]
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

The events received from the StbM have a fixed structure. For more details, please refer to [SWS_StbM_00284]

New chapter: StbM Use Case: Process time snapshot obtained from global time slave for diagnostics purposes

Scenario: a software-component provides a PPortPrototype onto which the global time slave pushes snapshots of time synchronization records. This data is typically used for diagnostic purposes.

[TPS_SWCT_0xxx1] StbM Use Case: Process time snapshot obtained from global time slave for diagnostics purposes

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
MeasurementNotification [1]
RoleBasedDataAssignment
N/A

RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

Note that in this case the SWC acts as a server, the StbM implements a client role!

[TPS_SWCT_0xxx2] Suffix used for the resulting name of the PortInterface for measurement notification
The suffix used for the resulting name of the PortInterface for the measurement notification (MeasurementNotification_TB_Name) shall be taken from the shortName of the applicable SwcServiceDependency.

New chapter: StbM Use Case: software-component represents a global time master

Scenario: a software-component implements the application-software part of the global time master role. For this purpose the software-component exposes an RPortPrototype that is supposed to be connected to the StbM service component.

[TPS_SWCT_0xxx3] StbM Use Case: software-component represents a global time master

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
GlobalTime_Master [1]
StatusNotification [0..1]
MeasurementNotification [0..1
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

[TPS_SWCT_0xxx4] Suffix used for the resulting name of the PortInterface for the global time master role
The suffix used for the resulting name of the PortInterface for the global time master role (GlobalTime_Master_Name) shall be taken from the shortName of the

applicable SwcServiceDependency.

New chapter: StbM Use Case: software-component represents a global time slave

Scenario: a software-component implements the application-software part of the global time slave role. For this purpose the software-component exposes an RPortPrototype that is supposed to be connected to the StbM service component.

[TPS_SWCT_0xxx5] StbM Use Case: software-component represents a global time slave

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
GlobalTime_Slave [1]
StatusNotification [0..1]
MeasurementNotification [0..1]
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

[TPS_SWCT_0xxx6] Suffix used for the resulting name of the PortInterface for the global time slave role
The suffix used for the resulting name of the PortInterface for the global time slave role (GlobalTime_Slave_Name) shall be taken from the shortName of the applicable SwcServiceDependency.
–Last change on issue 76867 comment 12–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.55 Specification Item TPS_SWCT_01741

Trace References:

none

Content:

The suffix used for the resulting name of the PortInterface for the measurement notification MeasurementNotification_{TB_Name} shall be taken from the Referrable.shortName of the applicable SwcServiceDependency.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76867: Role values in TPS_SWCT_01678 and TPS_SWCT_01679

  Problem description:

  TPS_SWCT_01678 and TPS_SWCT_01679 provide service use cases for the interaction of software-components with the StbM.

  The values of the role attribute of a RoleBasedPortAssignment are supposed to indicate the PortInterface used for the related PortPrototype.

  The problem is that none of the values of the role attribute correspond to existing PortInterfaces in the SWS StbM.

  In particular, the document assumes the existence of three ClientServerInterfaces:

  * StbM_TimeBaseValue
  * StbM_TimeBase_TriggerCustomer
  * StbM_TimeBase_StateNotification

  The same PortInterfaces are also mentioned in the EXP_VFB document (which consequently is added to the impact).

  The StbM actually provides these ClientServerInterfaces:

  * StbM_GlobalTime_Master
  * StbM_GlobalTime_Slave
  * StbM_StartTimer
  * StbM_TimeNotification
  * StbM_MesurementNotification

  Also, one SenderReceiverInterface is available:

  * StbM_StatusNotification

This more or less means that the service use cases need to be redefined such that the usage of the existing service interfaces is described.

Agreed solution:

VFB
===

Update R17.14 (11.2.1 Intra-ECU features):
A ComponentPrototype is requiring some StbM Interfaces (example: ClientServerInterface StartTimer or SenderReceiverInterface StatusNotification).

SWCT
====

Add note to chapter 13.10, above chapter 13.10.1:

The general idea behind the time synchronization concept is that the role of global time master and global time slave are partly implemented in the application software. For this purpose, the application software provides PortPrototypes typed by the standardized PortInterfaces GlobalTime_Master and GlobalTime_Slave.

In many cases both PortInterfaces will be used by the application software of one ECU. This means that the ECU is a global time slave on one domain and a global time master on another domain.

In terms of modeling, a given global time domain is represented by a SwcServiceDependency. If one software-component has to deal with different global time domains (e.g. because it represents a slave in one domain and a master in another) then the corresponding SwcInternalBehavior needs to define one SwcServiceDependency per global time domain.

Rework chapter 13.10.1:

StbM use case: start timer and potentially get notified about its expiration

Scenario: a software-component wants to wind up a timer in the StbM with a given expiration time. The software-component may want to receive a notification when the timer expires. In this case the following setup applies:

[TPS_SWCT_01678] StbM use case: start timer and potentially get notified about its expiration

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
StartTimer [1]
TimeNotification [0..1]
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

In this case the software-component needs to have an RPortPrototype typed by the ClientServerInterface StartTimer and (if applicable) a PPortPrototype typed by the ClientServerInterface TimeNotification.

Rework chapter 13.10.2:

StbM Use Case: Software-Components wants to get notifications of status changes

Scenario: a software-component wants to receive events whenever the status of the StbM changes. For this purpose, the software-component sports a sender/receiver RPortPrototype.

[TPS_SWCT_01679] StbM Use Case: Software-Components wants to get notifications of status changes

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
StatusNotification [1]
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

The events received from the StbM have a fixed structure. For more details, please refer to [SWS_StbM_00284]

New chapter: StbM Use Case: Process time snapshot obtained from global time slave for diagnostics purposes

Scenario: a software-component provides a PPortPrototype onto which the global time slave pushes snapshots of time synchronization records. This data is typically used for diagnostic purposes.

[TPS_SWCT_0xxx1] StbM Use Case: Process time snapshot obtained from global time slave for diagnostics purposes

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
MeasurementNotification [1]
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

Note that in this case the SWC acts as a server, the StbM implements a client role!

[TPS_SWCT_0xxx2] Suffix used for the resulting name of the PortInterface for measurement notification
The suffix used for the resulting name of the PortInterface for the measurement notification (MeasurementNotification_TB_Name) shall be taken from the shortName of the applicable SwcServiceDependency.

New chapter: StbM Use Case: software-component represents a global time master

Scenario: a software-component implements the application-software part of the global time master role. For this purpose the software-component exposes an RPortPrototype that is supposed to be connected to the StbM service component.

[TPS_SWCT_0xxx3] StbM Use Case: software-component represents a global time master

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
GlobalTime_Master [1]
StatusNotification [0..1]
MeasurementNotification [0..1
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

[TPS_SWCT_0xxx4] Suffix used for the resulting name of the PortInterface for the global time master role
The suffix used for the resulting name of the PortInterface for the global time master role (GlobalTime_Master_Name) shall be taken from the shortName of the applicable SwcServiceDependency.

New chapter: StbM Use Case: software-component represents a global time slave

Scenario: a software-component implements the application-software part of the global time slave role. For this purpose the software-component exposes an RPortPrototype that is supposed to be connected to the StbM service component.

[TPS_SWCT_0xxx5] StbM Use Case: software-component represents a global time slave

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
GlobalTime_Slave [1]
StatusNotification [0..1]
MeasurementNotification [0..1]
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups

N/A
()

[TPS_SWCT_0xxx6] Suffix used for the resulting name of the PortInterface for the global time slave role
The suffix used for the resulting name of the PortInterface for the global time slave role (GlobalTime_Slave_Name) shall be taken from the shortName of the applicable SwcServiceDependency.
–Last change on issue 76867 comment 12–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.56 Specification Item TPS_SWCT_01742

Trace References:

none

Content:

ServiceNeeds Kind : SyncTimeBaseMgrUserNeeds

RoleBasedPortAssignment valid roles:

- GlobalTime_Master [1]
- StatusNotification [0..1]
- MeasurementNotification [0..1]

RoleBasedDataAssignment N/A

RoleBasedDataTypeAssignment N/A

RepresentedPortGroups N/A

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76867: Role values in TPS_SWCT_01678 and TPS_SWCT_01679

  Problem description:

  TPS_SWCT_01678 and TPS_SWCT_01679 provide service use cases for the interaction of software-components with the StbM.

The values of the role attribute of a RoleBasedPortAssignment are supposed to indicate the PortInterface used for the related PortPrototype.

The problem is that none of the values of the role attribute correspond to existing PortInterfaces in the SWS StbM.

In particular, the document assumes the existence of three ClientServerInterfaces:

* StbM_TimeBaseValue
* StbM_TimeBase_TriggerCustomer
* StbM_TimeBase_StateNotification

The same PortInterfaces are also mentioned in the EXP_VFB document (which consequently is added to the impact).

The StbM actually provides these ClientServerInterfaces:

* StbM_GlobalTime_Master
* StbM_GlobalTime_Slave
* StbM_StartTimer
* StbM_TimeNotification
* StbM_MesurementNotification

Also, one SenderReceiverInterface is available:

* StbM_StatusNotification

This more or less means that the service use cases need to be redefined such that the usage of the existing service interfaces is described.

Agreed solution:

VFB
===

Update R17.14 (11.2.1 Intra-ECU features):
A ComponentPrototype is requiring some StbM Interfaces (example: ClientServerInterface StartTimer or SenderReceiverInterface StatusNotification).

SWCT
====

Add note to chapter 13.10, above chapter 13.10.1:

The general idea behind the time synchronization concept is that the role of global time master and global time slave are partly implemented in the application software. For this purpose, the application software provides PortPrototypes typed by the standardized PortInterfaces GlobalTime_Master and GlobalTime_Slave.

In many cases both PortInterfaces will be used by the application software of one ECU. This means that the ECU is a global time slave on one domain and a global time master on another domain.

In terms of modeling, a given global time domain is represented by a SwcServiceDependency. If one software-component has to deal with different global time domains (e.g. because it represents a slave in one domain and a master in another) then the corresponding SwcInternalBehavior needs to define one SwcServiceDependency per global time domain.

Rework chapter 13.10.1:

StbM use case: start timer and potentially get notified about its expiration

Scenario: a software-component wants to wind up a timer in the StbM with a given expiration time. The software-component may want to receive a notification when the timer expires. In this case the following setup applies:

[TPS_SWCT_01678] StbM use case: start timer and potentially get notified about its expiration

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
StartTimer [1]
TimeNotification [0..1]
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

In this case the software-component needs to have an RPortPrototype typed by the ClientServerInterface StartTimer and (if applicable) a PPortPrototype typed by the ClientServerInterface TimeNotification.

Rework chapter 13.10.2:

StbM Use Case: Software-Components wants to get notifications of status changes

Scenario: a software-component wants to receive events whenever the status of the StbM changes. For this purpose, the software-component sports a sender/receiver RPortPrototype.

[TPS_SWCT_01679] StbM Use Case: Software-Components wants to get notifications of status changes

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
StatusNotification [1]
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

The events received from the StbM have a fixed structure. For more details, please refer to [SWS_StbM_00284]

New chapter: StbM Use Case: Process time snapshot obtained from global time slave for diagnostics purposes

Scenario: a software-component provides a PPortPrototype onto which the global time slave pushes snapshots of time synchronization records. This data is typically used for diagnostic purposes.

[TPS_SWCT_0xxx1] StbM Use Case: Process time snapshot obtained from global time slave for diagnostics purposes

Service Needs kind SyncTimeBaseMgrUserNeeds

RoleBasedPortAssignment valid roles:
MeasurementNotification [1]
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

Note that in this case the SWC acts as a server, the StbM implements a client role!

[TPS_SWCT_0xxx2] Suffix used for the resulting name of the PortInterface for measurement notification
The suffix used for the resulting name of the PortInterface for the measurement notification (MeasurementNotification_TB_Name) shall be taken from the shortName of the applicable SwcServiceDependency.

New chapter: StbM Use Case: software-component represents a global time master

Scenario: a software-component implements the application-software part of the global time master role. For this purpose the software-component exposes an RPortPrototype that is supposed to be connected to the StbM service component.

[TPS_SWCT_0xxx3] StbM Use Case: software-component represents a global time master

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
GlobalTime_Master [1]
StatusNotification [0..1]
MeasurementNotification [0..1
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

[TPS_SWCT_0xxx4] Suffix used for the resulting name of the PortInterface for the global time master role
The suffix used for the resulting name of the PortInterface for the global time master role (GlobalTime_Master_Name) shall be taken from the shortName of the applicable SwcServiceDependency.

New chapter: StbM Use Case: software-component represents a global time slave

Scenario: a software-component implements the application-software part of the global time slave role. For this purpose the software-component exposes an RPortPrototype that is supposed to be connected to the StbM service component.

[TPS_SWCT_0xxx5] StbM Use Case: software-component represents a global time slave

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
GlobalTime_Slave [1]
StatusNotification [0..1]
MeasurementNotification [0..1]
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

[TPS_SWCT_0xxx6] Suffix used for the resulting name of the PortInterface for the global time slave role
The suffix used for the resulting name of the PortInterface for the global time slave role (GlobalTime_Slave_Name) shall be taken from the shortName of the applicable SwcServiceDependency.
–Last change on issue 76867 comment 12–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.57 Specification Item TPS_SWCT_01743

Trace References:

none

Content:

The suffix used for the resulting name of the PortInterface for the global time master role GlobalTime_Master_{Name} shall be taken from the Referrable.shortName of the applicable SwcServiceDependency.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76867: Role values in TPS_SWCT_01678 and TPS_SWCT_01679

  Problem description:

  TPS_SWCT_01678 and TPS_SWCT_01679 provide service use cases for the interaction of software-components with the StbM.

  The values of the role attribute of a RoleBasedPortAssignment are supposed to indicate the PortInterface used for the related PortPrototype.

  The problem is that none of the values of the role attribute correspond to existing PortInterfaces in the SWS StbM.

  In particular, the document assumes the existence of three ClientServerInterfaces:

  * StbM_TimeBaseValue
  * StbM_TimeBase_TriggerCustomer
  * StbM_TimeBase_StateNotification

  The same PortInterfaces are also mentioned in the EXP_VFB document (which consequently is added to the impact).

  The StbM actually provides these ClientServerInterfaces:

  * StbM_GlobalTime_Master
  * StbM_GlobalTime_Slave
  * StbM_StartTimer
  * StbM_TimeNotification
  * StbM_MesurementNotification

Also, one SenderReceiverInterface is available:

* StbM_StatusNotification

This more or less means that the service use cases need to be redefined such that the usage of the existing service interfaces is described.

Agreed solution:

VFB
===

Update R17.14 (11.2.1 Intra-ECU features):
A ComponentPrototype is requiring some StbM Interfaces (example: ClientServerInterface StartTimer or SenderReceiverInterface StatusNotification).

SWCT
====

Add note to chapter 13.10, above chapter 13.10.1:

The general idea behind the time synchronization concept is that the role of global time master and global time slave are partly implemented in the application software. For this purpose, the application software provides PortPrototypes typed by the standardized PortInterfaces GlobalTime_Master and GlobalTime_Slave.

In many cases both PortInterfaces will be used by the application software of one ECU. This means that the ECU is a global time slave on one domain and a global time master on another domain.

In terms of modeling, a given global time domain is represented by a SwcServiceDependency. If one software-component has to deal with different global time domains (e.g. because it represents a slave in one domain and a master in another) then the corresponding SwcInternalBehavior needs to define one SwcServiceDependency per global time domain.

Rework chapter 13.10.1:

StbM use case: start timer and potentially get notified about its expiration

Scenario: a software-component wants to wind up a timer in the StbM with a given expiration time. The software-component may want to receive a notification when the timer expires. In this case the following setup applies:

[TPS_SWCT_01678] StbM use case: start timer and potentially get notified about its expiration

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
StartTimer [1]
TimeNotification [0..1]
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

In this case the software-component needs to have an RPortPrototype typed by the ClientServerInterface StartTimer and (if applicable) a PPortPrototype typed by the ClientServerInterface TimeNotification.

Rework chapter 13.10.2:

StbM Use Case: Software-Components wants to get notifications of status changes

Scenario: a software-component wants to receive events whenever the status of the StbM changes. For this purpose, the software-component sports a sender/receiver RPortPrototype.

[TPS_SWCT_01679] StbM Use Case: Software-Components wants to get notifications of status changes

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
StatusNotification [1]
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups

N/A
()

The events received from the StbM have a fixed structure. For more details, please refer to [SWS_StbM_00284]

New chapter: StbM Use Case: Process time snapshot obtained from global time slave for diagnostics purposes

Scenario: a software-component provides a PPortPrototype onto which the global time slave pushes snapshots of time synchronization records. This data is typically used for diagnostic purposes.

[TPS_SWCT_0xxx1] StbM Use Case: Process time snapshot obtained from global time slave for diagnostics purposes

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
MeasurementNotification [1]
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

Note that in this case the SWC acts as a server, the StbM implements a client role!

[TPS_SWCT_0xxx2] Suffix used for the resulting name of the PortInterface for measurement notification
The suffix used for the resulting name of the PortInterface for the measurement notification (MeasurementNotification_TB_Name) shall be taken from the shortName of the applicable SwcServiceDependency.

New chapter: StbM Use Case: software-component represents a global time master

Scenario: a software-component implements the application-software part of the global time master role. For this purpose the software-component exposes an RPortPrototype that is supposed to be connected to the StbM service component.

[TPS_SWCT_0xxx3] StbM Use Case: software-component represents a global time master

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
GlobalTime_Master [1]
StatusNotification [0..1]
MeasurementNotification [0..1
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

[TPS_SWCT_0xxx4] Suffix used for the resulting name of the PortInterface for the global time master role
The suffix used for the resulting name of the PortInterface for the global time master role (GlobalTime_Master_Name) shall be taken from the shortName of the applicable SwcServiceDependency.

New chapter: StbM Use Case: software-component represents a global time slave

Scenario: a software-component implements the application-software part of the global time slave role. For this purpose the software-component exposes an RPortPrototype that is supposed to be connected to the StbM service component.

[TPS_SWCT_0xxx5] StbM Use Case: software-component represents a global time slave

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
GlobalTime_Slave [1]
StatusNotification [0..1]
MeasurementNotification [0..1]
RoleBasedDataAssignment

N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

[TPS_SWCT_0xxx6] Suffix used for the resulting name of the PortInterface for the global time slave role
The suffix used for the resulting name of the PortInterface for the global time slave role (GlobalTime_Slave_Name) shall be taken from the shortName of the applicable SwcServiceDependency.
–Last change on issue 76867 comment 12–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.58 Specification Item TPS_SWCT_01744

Trace References:

none

Content:

ServiceNeeds Kind : SyncTimeBaseMgrUserNeeds

RoleBasedPortAssignment valid roles:

- GlobalTime_Slave [1]
- StatusNotification [0..1]
- MeasurementNotification [0..1]

RoleBasedDataAssignment N/A

RoleBasedDataTypeAssignment N/A

RepresentedPortGroups N/A

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76867: Role values in TPS_SWCT_01678 and TPS_SWCT_01679

  Problem description:

TPS_SWCT_01678 and TPS_SWCT_01679 provide service use cases for the interaction of software-components with the StbM.

The values of the role attribute of a RoleBasedPortAssignment are supposed to indicate the PortInterface used for the related PortPrototype.

The problem is that none of the values of the role attribute correspond to existing PortInterfaces in the SWS StbM.

In particular, the document assumes the existence of three ClientServerInterfaces:

* StbM_TimeBaseValue
* StbM_TimeBase_TriggerCustomer
* StbM_TimeBase_StateNotification

The same PortInterfaces are also mentioned in the EXP_VFB document (which consequently is added to the impact).

The StbM actually provides these ClientServerInterfaces:

* StbM_GlobalTime_Master
* StbM_GlobalTime_Slave
* StbM_StartTimer
* StbM_TimeNotification
* StbM_MesurementNotification

Also, one SenderReceiverInterface is available:

* StbM_StatusNotification

This more or less means that the service use cases need to be redefined such that the usage of the existing service interfaces is described.

Agreed solution:

VFB
===

Update R17.14 (11.2.1 Intra-ECU features):
A ComponentPrototype is requiring some StbM Interfaces (example: ClientServerInterface StartTimer or SenderReceiverInterface StatusNotification).

SWCT

====

Add note to chapter 13.10, above chapter 13.10.1:

The general idea behind the time synchronization concept is that the role of global time master and global time slave are partly implemented in the application software. For this purpose, the application software provides PortPrototypes typed by the standardized PortInterfaces GlobalTime_Master and GlobalTime_Slave.

In many cases both PortInterfaces will be used by the application software of one ECU. This means that the ECU is a global time slave on one domain and a global time master on another domain.

In terms of modeling, a given global time domain is represented by a SwcServiceDependency. If one software-component has to deal with different global time domains (e.g. because it represents a slave in one domain and a master in another) then the corresponding SwcInternalBehavior needs to define one SwcServiceDependency per global time domain.

Rework chapter 13.10.1:

StbM use case: start timer and potentially get notified about its expiration

Scenario: a software-component wants to wind up a timer in the StbM with a given expiration time. The software-component may want to receive a notification when the timer expires. In this case the following setup applies:

[TPS_SWCT_01678] StbM use case: start timer and potentially get notified about its expiration

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
StartTimer [1]
TimeNotification [0..1]
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

In this case the software-component needs to have an RPortPrototype typed by the ClientServerInterface StartTimer and (if applicable) a PPortPrototype typed by the ClientServerInterface TimeNotification.

Rework chapter 13.10.2:

StbM Use Case: Software-Components wants to get notifications of status changes

Scenario: a software-component wants to receive events whenever the status of the StbM changes. For this purpose, the software-component sports a sender/receiver RPortPrototype.

[TPS_SWCT_01679] StbM Use Case: Software-Components wants to get notifications of status changes

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
StatusNotification [1]
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

The events received from the StbM have a fixed structure. For more details, please refer to [SWS_StbM_00284]

New chapter: StbM Use Case: Process time snapshot obtained from global time slave for diagnostics purposes

Scenario: a software-component provides a PPortPrototype onto which the global time slave pushes snapshots of time synchronization records. This data is typically used for diagnostic purposes.

[TPS_SWCT_0xxx1] StbM Use Case: Process time snapshot obtained from global time slave for diagnostics purposes

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
MeasurementNotification [1]
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

Note that in this case the SWC acts as a server, the StbM implements a client role!

[TPS_SWCT_0xxx2] Suffix used for the resulting name of the PortInterface for measurement notification
The suffix used for the resulting name of the PortInterface for the measurement notification (MeasurementNotification_TB_Name) shall be taken from the shortName of the applicable SwcServiceDependency.

New chapter: StbM Use Case: software-component represents a global time master

Scenario: a software-component implements the application-software part of the global time master role. For this purpose the software-component exposes an RPortPrototype that is supposed to be connected to the StbM service component.

[TPS_SWCT_0xxx3] StbM Use Case: software-component represents a global time master

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
GlobalTime_Master [1]
StatusNotification [0..1]
MeasurementNotification [0..1
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups

N/A
()

[TPS_SWCT_0xxx4] Suffix used for the resulting name of the PortInterface for the global time master role
The suffix used for the resulting name of the PortInterface for the global time master role (GlobalTime_Master_Name) shall be taken from the shortName of the applicable SwcServiceDependency.

New chapter: StbM Use Case: software-component represents a global time slave

Scenario: a software-component implements the application-software part of the global time slave role. For this purpose the software-component exposes an RPortPrototype that is supposed to be connected to the StbM service component.

[TPS_SWCT_0xxx5] StbM Use Case: software-component represents a global time slave

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
GlobalTime_Slave [1]
StatusNotification [0..1]
MeasurementNotification [0..1]
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

[TPS_SWCT_0xxx6] Suffix used for the resulting name of the PortInterface for the global time slave role
The suffix used for the resulting name of the PortInterface for the global time slave role (GlobalTime_Slave_Name) shall be taken from the shortName of the applicable SwcServiceDependency.
–Last change on issue 76867 comment 12–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.59 Specification Item TPS_SWCT_01745

Trace References:

none

Content:

The suffix used for the resulting name of the PortInterface for the global time slave role GlobalTime_Slave_{Name} shall be taken from the Referrable.shortName of the applicable SwcServiceDependency.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #76867: Role values in TPS_SWCT_01678 and TPS_SWCT_01679

  Problem description:

  TPS_SWCT_01678 and TPS_SWCT_01679 provide service use cases for the interaction of software-components with the StbM.

  The values of the role attribute of a RoleBasedPortAssignment are supposed to indicate the PortInterface used for the related PortPrototype.

  The problem is that none of the values of the role attribute correspond to existing PortInterfaces in the SWS StbM.

  In particular, the document assumes the existence of three ClientServerInterfaces:

  * StbM_TimeBaseValue
  * StbM_TimeBase_TriggerCustomer
  * StbM_TimeBase_StateNotification

  The same PortInterfaces are also mentioned in the EXP_VFB document (which consequently is added to the impact).

  The StbM actually provides these ClientServerInterfaces:

  * StbM_GlobalTime_Master
  * StbM_GlobalTime_Slave

* StbM_StartTimer
* StbM_TimeNotification
* StbM_MesurementNotification

Also, one SenderReceiverInterface is available:

* StbM_StatusNotification

This more or less means that the service use cases need to be redefined such that the usage of the existing service interfaces is described.

Agreed solution:

VFB
===

Update R17.14 (11.2.1 Intra-ECU features):
A ComponentPrototype is requiring some StbM Interfaces (example: ClientServerInterface StartTimer or SenderReceiverInterface StatusNotification).

SWCT
====

Add note to chapter 13.10, above chapter 13.10.1:

The general idea behind the time synchronization concept is that the role of global time master and global time slave are partly implemented in the application software. For this purpose, the application software provides PortPrototypes typed by the standardized PortInterfaces GlobalTime_Master and GlobalTime_Slave.

In many cases both PortInterfaces will be used by the application software of one ECU. This means that the ECU is a global time slave on one domain and a global time master on another domain.

In terms of modeling, a given global time domain is represented by a SwcServiceDependency. If one software-component has to deal with different global time domains (e.g. because it represents a slave in one domain and a master in another) then the corresponding SwcInternalBehavior needs to define one SwcServiceDependency per global time domain.

Rework chapter 13.10.1:

StbM use case: start timer and potentially get notified about its expiration

Scenario: a software-component wants to wind up a timer in the StbM with a given expiration time. The software-component may want to receive a notification when the timer expires. In this case the following setup applies:

[TPS_SWCT_01678] StbM use case: start timer and potentially get notified about its expiration

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
StartTimer [1]
TimeNotification [0..1]
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

In this case the software-component needs to have an RPortPrototype typed by the ClientServerInterface StartTimer and (if applicable) a PPortPrototype typed by the ClientServerInterface TimeNotification.

Rework chapter 13.10.2:

StbM Use Case: Software-Components wants to get notifications of status changes

Scenario: a software-component wants to receive events whenever the status of the StbM changes. For this purpose, the software-component sports a sender/receiver RPortPrototype.

[TPS_SWCT_01679] StbM Use Case: Software-Components wants to get notifications of status changes

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
StatusNotification [1]
RoleBasedDataAssignment

N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

The events received from the StbM have a fixed structure. For more details, please refer to [SWS_StbM_00284]

New chapter: StbM Use Case: Process time snapshot obtained from global time slave for diagnostics purposes

Scenario: a software-component provides a PPortPrototype onto which the global time slave pushes snapshots of time synchronization records. This data is typically used for diagnostic purposes.

[TPS_SWCT_0xxx1] StbM Use Case: Process time snapshot obtained from global time slave for diagnostics purposes

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
MeasurementNotification [1]
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

Note that in this case the SWC acts as a server, the StbM implements a client role!

[TPS_SWCT_0xxx2] Suffix used for the resulting name of the PortInterface for measurement notification
The suffix used for the resulting name of the PortInterface for the measurement notification (MeasurementNotification_TB_Name) shall be taken from the shortName of the applicable SwcServiceDependency.

New chapter: StbM Use Case: software-component represents a global time master

Scenario: a software-component implements the application-software part of the global time master role. For this purpose the software-component exposes an RPortPrototype that is supposed to be connected to the StbM service component.

[TPS_SWCT_0xxx3] StbM Use Case: software-component represents a global time master

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:
GlobalTime_Master [1]
StatusNotification [0..1]
MeasurementNotification [0..1
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

[TPS_SWCT_0xxx4] Suffix used for the resulting name of the PortInterface for the global time master role
The suffix used for the resulting name of the PortInterface for the global time master role (GlobalTime_Master_Name) shall be taken from the shortName of the applicable SwcServiceDependency.

New chapter: StbM Use Case: software-component represents a global time slave

Scenario: a software-component implements the application-software part of the global time slave role. For this purpose the software-component exposes an RPortPrototype that is supposed to be connected to the StbM service component.

[TPS_SWCT_0xxx5] StbM Use Case: software-component represents a global time slave

Service Needs kind SyncTimeBaseMgrUserNeeds
RoleBasedPortAssignment valid roles:

GlobalTime_Slave [1]
StatusNotification [0..1]
MeasurementNotification [0..1]
RoleBasedDataAssignment
N/A
RoleBasedDataTypeAssignment
N/A
RepresentedPortGroups
N/A
()

[TPS_SWCT_0xxx6] Suffix used for the resulting name of the PortInterface for the global time slave role
The suffix used for the resulting name of the PortInterface for the global time slave role (GlobalTime_Slave_Name) shall be taken from the shortName of the applicable SwcServiceDependency.
–Last change on issue 76867 comment 12–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.60 Specification Item TPS_SWCT_01746

Trace References:

RS_SWCT_03310, RS_SWCT_03190

Content:

ServiceNeeds kind FurtherActionByteNeeds

RoleBasedPortAssignment valid roles:

- CallbackGetFurtherActionByte [1]

RoleBasedDataAssignment N/A

RepresentedPortGroups N/A

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #74849: [DoIP] Further action code values (0x11 to 0xFF) missing

  Problem description:

Further action code values (0x11 to 0xFF) that are defined in ISO 13400-2 missing in Autosar standard. The further action byte is used in vehicle identification/vehicle announcement.

ISO 13400-2 mentions range (0x11 to 0xFF) of "further action byte" value in "Table 20 Definition of further action code values" for OEM specific use but this is not mentioned in Autosar standard.

0x11 to 0xFF Available for additional OEM-specific use. optional

Please add a configurable parameter (i.e. DoIPGetFurtherActionByteCallback) under DoIPGeneral as DoIPGeneral/DoIPGetFurtherActionByteCallback

which should be a string literal. With lower multiplicity 0. If configured, this will be the name of the callback to get the OEM specific further action byte to be used in vehicle identification/announcement message.

Agreed solution:

1- Changes in Chapter 7:

Table 6: Vehicle identification response/vehicle announcement message payload data
Further action required should be changed to Further action or Further action byte.

7.3.2.2.4 Vehicle Identification response/vehicle announcement (payload type 0x0004)
Add [SWS_DoIP_00xxx]
The Further action byte of a vehicle identification response/vehicle announcement message shall contain the 1 Byte value retrieved by a call to the configured DoIPFurtherActionByteCallback (if configured, for the signature see <User>_DoIPGetFurtherActionByteCallback, SWS_DoIP_00xxx). If the function returns E_OK, the Further action byte shall be set to the retrieved value of FurtherActionByte. If the function returns E_NOT_OK, the Further action byte shall be set according to [SWS_DoIP_00082], [SWS_DoIP_00083] or [SWS_DoIP_00084].
(SRS_Eth_00026)

2- Changes in Chapter 8

8.1 Imported types
Add [SWS_DoIP_00xxx]
Name DoIP_FurtherActionByteType
Kind Type
Derived from uint8

Description Used to get the OEM specific Further Action Byte for the DoIP vehicle identification response/vehicle announcement.
Range
0x11-0xFF 0x11-0xFF Available for additional OEM-specific use
Variation –
()

8.6.3 Configurable interfaces
8.6.3.6 <User>_DoIPGetFurtherActionByteCallback
Add [SWS_DoIP_00xxx]
Service name: <User>_ DoIPGetFurtherActionByteCallback
Syntax: Std_ReturnType <User>_ DoIPGetFurtherActionByteCallback (DoIP_FurtherActionByteType* FurtherActionByte)
Service ID[hex]: 0x00
Sync/Async: Synchronous
Reentrancy: Don't care
Parameters (in): None
Parameters (inout): None
Parameters (out): FurtherActionByte
Pointer containing the information of the FurtherActionByte. Only valid if the return value equals E_OK.
Return value: Std_ReturnType
E_OK: FurtherActionByte contains valid information
E_NOT_OK: FurtherActionByte contains no valid information
Description: Callback function to get the OEM specific Further Action Byte for the DoIP vehicle identification response/vehicle announcement.
(SRS_Eth_00026)

8.6.4 DoIP Service Component
The DoIP Service Component shall provide the interface CallbackGetFurtherActionByte as described below to request the value of the OEM specific Further Action Byte for the DoIP vehicle identification response/vehicle announcement.

Name CallbackGetFurtherActionByte
Comment –
IsService true
Variation ecuc(DoIP/DoIPGeneral/DoIPFurtherActionByteCallback/DoIPFurtherActionByteDirect)
== NULL
Possible Errors
0 E_OK
1 E_NOT_OK

Operations

GetFurtherActionByte
Comments –
Variation –
Parameters FurtherActionByte
Comment –
Type DoIP_FurtherActionByteType
Variation –
Direction OUT
Possible Errors
E_OK Operation successful
E_NOT_OK –

The DoIP Service Component shall be equipped with a service port as described below to request the value of the Further Action Byte for DoIP diagnostic vehicle identification response/vehicle announcement.

Add [SWS_DoIP_00xxx]
Name CBGetfurtherActionByte
Kind RequiredPort
Interface CallbackGetFurtherActionByte
Description –
Variation ecuc(DoIP/DoIPGeneral/DoIPFurtherActionByteCallback/DoIPFurtherActionByteDirect)
== NULL

3- Changes in Chapter 10

10.2.3 DoIPGeneral
Included Containers
Container Name: DoIPFurtherActionByteCallback
Multiplicity: 0..1
Scope / Dependency: This container describes the Callbackfunction to get the Further Action byte. If this container is not configured no Callbackfunction will be used.
If the DoIPFurtherActionByteDirect parameter is not present, the DoIP module will use an RPort of ServiceInterface CallbackGetFurtherActionByte with the name "CBGetFurtherActionByte".

10.2.x DoIPFurtherActionByteCallback
SWS Item: ECUC_DoIP_000xx
Container Name: DoIPFurtherActionByteCallback
Description: This container describes the Callbackfunction to get the Further Action byte. This container shall always be present. If the DoIPFurtherActionByteDirect parameter is not present, the DoIP module will use an RPort of ServiceInterface

CallbackGetPowerMode with the name "CBGetFurtherActionByte".

Add Configuration Parameters as:

SWS Item: ECUC_DoIP_00xxx
Name: DoIPFurtherActionByteDirect
Description: Direct C Callback function to get the OEM specific Further Action Byte for the DoIP vehicle identification response/vehicle announcement. If the DoIPFurtherActionByteDirect parameter is present, the DoIP module will not use an RPort of ServiceInterface CBGetFurtherActionByte but will call the configured function.
Multiplicity 0..1
Type EcucFunctionNameDef
Default value –
maxLength –
minLength –
regularExpression
ConfigurationClass
Pre-compile time X VARIANT-PRE-COMPILE
Link time X VARIANT-LINK-TIME, VARIANT-POST-BUILD
Post-build time –
Scope / Dependency scope: local
No Included Containers

SWCT
====

Add new chapter below 13.8.6:

13.8.6.x DoIP Service Use Case: Atomic Software-Component provides the further action byte to the DoIP Service Component.

Scenario: An AtomicSoftwareComponent provides the "further action byte" used in vehicle identification/announcement message.

[TPS_SWCT_0xxxx] DoIP Service Use Case: Atomic Software-Component provides the further action byte
ServiceNeeds kind DiagnosticComponentNeeds
ServiceNeedsKind: FurtherActionByteNeeds
RoleBasedPortAssignment valid roles:
CallbackGetFurtherActionByte[1]
RoleBasedDataAssignment
N/A

RepresentedPortGroups
N/A
c(RS_SWCT_03310, RS_SWCT_03190)

Add new subclass of DoIpServiceNeeds named FurtherActionByteNeeds. Description: The FurtherActionByteNeeds indicates that the software-component is able to provide the "further action byte" to the DoIp Service Component."
–Last change on issue 74849 comment 10–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.61 Specification Item TPS_SWCT_01747

Trace References:

none

Content:

A fix axis shall be modeled as an SwCalprmAxis with attribute Identifiable.category set to the value FIX_AXISFIX_AXIS.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77620: Clarifiy description of FIX_AXIS

  Problem description:

  The AUTOSAR Meta Model is prepared for the description of FIX_AXIS but the SWC-T does not provide further information how to use it.

  Agreed solution:

  SWCT
  ====

  Create new chapter 5.4.4.1 Specification of fix axes

  In most cases the axes of a curve or map are accessible to a calibration software and it is possible to calibrate axes points and their corresponding values. There are cases, however, where axes are intentionally declared as fix and where no intention exists to change the properties of the axis ever[fn1].

These axes are also known as fix axes.

[TPS_SWCT_0xxx] Value of category for fix axis
A fix axis shall be modeled as an SwCalprmAxis with attribute category set to the value FIX_AXIS.

[TPS_SWCT_0xxx1] Sub-categories of fix axes
There are different sub-categories of fix axes:
* Fix axis where the distance between axis points can be computed according to a standardized algorithm. In this case, fix axes of arbitrary length can be described by feeding three arguments defined in the context of the axis description into the axis algorithm. Consequently, the memory footprint of different fix axis of this category is literally identical, independently of the number of axis points.
The following variations exist:
- Subcategory PAR, i.e. category = FIX_AXIS_PAR: the axis is created out of a starting value and a shift that creates further axis points as using a power-of-two algorithm. The details can be found in [27].
- Subcategory PAR_DIST, i.e. category = FIX_AXIS_PAR_DIST: the axis is created out of a starting value and an offset that adds further axis points with the distance given by offset. The details can be found in [27].
* Fix axis where the axis points are defined as a list of values directly in the axis definition. This variety boils down to
- Subcategory PAR_LIST, i.e. category = FIX_AXIS_PAR_LIST: the axis is created out of a list of numerical values that represent the axis points. The details can be found in [27].
These values of category shall be used for SwAxisType.

In other words, since the values that define a fix axis are modeled directly in the axis model there is no use case for assigning an initial value to a fix axis.

However, the calibration software may have still have access to axis points and values of the fix axis if these properties are specified in an A2L file. For this purpose McDataInstance needs to be set up properly. The details are explained in [7].

The modeling of of an axis of category FIX_AXIS_PAR is sketched in the following example model.

<insert attachment 4552>

The modeling of of an axis of category FIX_AXIS_PAR_DIST is sketched in the following example model.

<insert attachment 4553>

The modeling of of an axis of category FIX_AXIS_PAR_LIST is sketched in the following example model.

<insert attachment 4555>

Please note that the axis points and values of a fix axis are defined in the definition of the fix axis itself and therefore any initial value asigned to a fix axis would be ignored anyway. This might lead to confusion such that the initial value does not make it into the software. In order to avoid such confusion AUTOSAR does not support the definition of an initial value for a fix axis. This regulation is reflected in the existence of constr_xxx1.

[constr_0xxx1] No initialization for fix axis
an ApplicationValueSpecifciation taken to initialize an ApplicationPrimitiveDataType that contains a fix axis shall not contain initial values for the axis index of the fix axis inside the ApplicationPrimitiveDataType.

Add table:

category of swAxisType category of SwGenericAxisParamType Multiplicity
FIX_AXIS_PAR OFFSET 1
SHIFT 1
FIX_AXIS_PAR_DIST OFFSET 1
DISTANCE 1
FIX_AXIS_PAR_LIST LIST 1

Add constraint that references the table.

Changes in the meta-model:

* Remove the mention of FIX_AXIS from the description of RuleBasedAxisCont.category (because fix axis are not initialized by means of a ValueSpecifiction)

* Remove the mention of FIX_AXIS from the description of SwAxisCont.category (because fix axis are not initialized by means of a ValueSpecifiction)

* make SwAxisGenericParam.vf ordered.

[fn1]: Typically, a calibration software does not have the ability to manipulate (or even inspect) the axis' properties by inspecting the ECU's memory.

[27]: ASAM MCD 2MC ASAP2 Interface Specification

[7]: Basic Software Module Description Template

GenBlprSupplement
=================

3.2.3 Record Layout: FixIntCur

This chapter describes the record layout for a curve with fixed axis points. Fixed axis exist in three categories: FIX_AXIS_PAR, FIX_AXIS_PAR_DIST and FIX_AXIS_PAR_LIST, see [TPS_SWCT_0xxx1].

The number of sampling points (Nx), the Offset, the shift and the distance values are represented in the following chapters by these logical views:

<insert figure with three representations>

These values are not defined inside SwRecordLayout(s) with fixed axis points.

Logical view:

The figure 3.9 illustrates the logical view of the SwRecordLayout FixIntCur. The SwRecordLayoutGroup with the shortLabel Val is shown in the lower part. Its elements are indexed by virtual [AXIS 1] which is fixed and of category FIX_AXIS_PAR and not defined inside this SwRecordLayout.

3.3.6 Record Layout: FixIntMap

This chapter describes the record layout for a map with fixed axis points. Fixed axis exist in three categories: FIX_AXIS_PAR, FIX_AXIS_PAR_DIST and FIX_AXIS_PAR_LIST, see [TPS_SWCT_xxxxx].

The number of sampling points (Nx, Ny), the Offset, the shift and the distance values are represented in the following chapters by these logical views:

<insert figure with three representations>

These values are not defined inside SwRecordLayout(s) with fixed axis points.

Logical view:
The figure 3.27 illustrates the logical view of the SwRecordLayout FixIntMap. The SwRecordLayoutGroup with the shortLabel Val is shown in the lower part. Its elements are indexed by [AXIS 2, AXIS 1] from value (AXIS 2: = 1, AXIS 1: = 1) to value (AXIS 2: = -1, AXIS 1: = -1) there -1 gives the last value.
–Last change on issue 77620 comment 26–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.62 Specification Item TPS_SWCT_01748

Trace References:

none

Content:

There are different sub-categories of fix axes:

- Fix axis where the distance between axis points can be computed according to a standardized algorithm.

  In this case, fix axes of arbitrary length can be described by feeding three arguments defined in the context of the axis description into the axis algorithm.

  Consequently, the memory footprint of different fix axis of this Identifiable.category is literally identical, independently of the number of axis points.

  The following variations exist:

  - Subcategory , i.e. Identifiable.category = FIX_AXIS_PARFIX_AXIS_PAR: the axis is created out of a starting value and a shift that creates further axis points as using a power-of-two algorithm. The details can be found in ASAM_2d_MCD_2d_2MC_2d_ASAP2.
  - Subcategory , i.e. Identifiable.category = FIX_AXIS_PAR_DISTFIX_AXIS_PAR_DIST: the axis is created out of a starting value and an offset that adds further axis points with the distance given by offset. The details can be found in ASAM_2d_MCD_2d_2MC_2d_ASAP2.

- Fix axis where the axis points are defined as a list of values directly in the axis definition. This variety boils down to

  - Subcategory , i.e. Identifiable.category = FIX_AXIS_PAR_LISTFIX_AXIS_PAR_LIST: the axis is created out of a list of numerical values that represent the axis points. The details can be found in ASAM_2d_MCD_2d_2MC_2d_ASAP2.

These values of Identifiable.category shall be used for SwAxisType.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77620: Clarifiy description of FIX_AXIS

  Problem description:

  The AUTOSAR Meta Model is prepared for the description of FIX_AXIS but the SWC-T does not provide further information how to use it.

  Agreed solution:

  SWCT
  ====

  Create new chapter 5.4.4.1 Specification of fix axes

  In most cases the axes of a curve or map are accessible to a calibration software and it is possible to calibrate axes points and their corresponding values. There are cases, however, where axes are intentionally declared as fix and where no intention exists to change the properties of the axis ever[fn1].

  These axes are also known as fix axes.

  [TPS_SWCT_0xxx] Value of category for fix axis
  A fix axis shall be modeled as an SwCalprmAxis with attribute category set to the value FIX_AXIS.

  [TPS_SWCT_0xxx1] Sub-categories of fix axes
  There are different sub-categories of fix axes:
  * Fix axis where the distance between axis points can be computed according to a standardized algorithm. In this case, fix axes of arbitrary length can be described by feeding three arguments defined in the context of the axis description into the axis algorithm. Consequently, the memory footprint of different fix axis of this category is literally identical, independently of the number of axis points.
  The following variations exist:
  - Subcategory PAR, i.e. category = FIX_AXIS_PAR: the axis is created out of a starting value and a shift that creates further axis points as using a power-of-two algorithm. The details can be found in [27].

- Subcategory PAR_DIST, i.e. category = FIX_AXIS_PAR_DIST: the axis is created out of a starting value and an offset that adds further axis points with the distance given by offset. The details can be found in [27].
* Fix axis where the axis points are defined as a list of values directly in the axis definition. This variety boils down to
- Subcategory PAR_LIST, i.e. category = FIX_AXIS_PAR_LIST: the axis is created out of a list of numerical values that represent the axis points. The details can be found in [27].
These values of category shall be used for SwAxisType.

In other words, since the values that define a fix axis are modeled directly in the axis model there is no use case for assigning an initial value to a fix axis.

However, the calibration software may have still have access to axis points and values of the fix axis if these properties are specified in an A2L file. For this purpose McDataInstance needs to be set up properly. The details are explained in [7].

The modeling of of an axis of category FIX_AXIS_PAR is sketched in the following example model.

<insert attachment 4552>

The modeling of of an axis of category FIX_AXIS_PAR_DIST is sketched in the following example model.

<insert attachment 4553>

The modeling of of an axis of category FIX_AXIS_PAR_LIST is sketched in the following example model.

<insert attachment 4555>

Please note that the axis points and values of a fix axis are defined in the definition of the fix axis itself and therefore any initial value asigned to a fix axis would be ignored anyway. This might lead to confusion such that the initial value does not make it into the software. In order to avoid such confusion AUTOSAR does not support the definition of an initial value for a fix axis. This regulation is reflected in the existence of constr_xxx1.

[constr_0xxx1] No initialization for fix axis
an ApplicationValueSpecifciation taken to initialize an ApplicationPrimitiveDataType

that contains a fix axis shall not contain initial values for the axis index of the fix axis inside the ApplicationPrimitiveDataType.

Add table:

category of swAxisType category of SwGenericAxisParamType Multiplicity
FIX_AXIS_PAR OFFSET 1
SHIFT 1
FIX_AXIS_PAR_DIST OFFSET 1
DISTANCE 1
FIX_AXIS_PAR_LIST LIST 1

Add constraint that references the table.

Changes in the meta-model:

* Remove the mention of FIX_AXIS from the description of RuleBasedAxisCont.category (because fix axis are not initialized by means of a ValueSpecifiction)

* Remove the mention of FIX_AXIS from the description of SwAxisCont.category (because fix axis are not initialized by means of a ValueSpecifiction)

* make SwAxisGenericParam.vf ordered.

[fn1]: Typically, a calibration software does not have the ability to manipulate (or even inspect) the axis' properties by inspecting the ECU's memory.

[27]: ASAM MCD 2MC ASAP2 Interface Specification

[7]: Basic Software Module Description Template

GenBlprSupplement
=================

3.2.3 Record Layout: FixIntCur

This chapter describes the record layout for a curve with fixed axis points. Fixed axis exist in three categories: FIX_AXIS_PAR, FIX_AXIS_PAR_DIST and FIX_AXIS_PAR_LIST, see [TPS_SWCT_0xxx1].

The number of sampling points (Nx), the Offset, the shift and the distance values are represented in the following chapters by these logical views:

<insert figure with three representations>

These values are not defined inside SwRecordLayout(s) with fixed axis points.

Logical view:

The figure 3.9 illustrates the logical view of the SwRecordLayout FixIntCur. The SwRecordLayoutGroup with the shortLabel Val is shown in the lower part. Its elements are indexed by virtual [AXIS 1] which is fixed and of category FIX_AXIS_PAR and not defined inside this SwRecordLayout.

3.3.6 Record Layout: FixIntMap

This chapter describes the record layout for a map with fixed axis points. Fixed axis exist in three categories: FIX_AXIS_PAR, FIX_AXIS_PAR_DIST and FIX_AXIS_PAR_LIST, see [TPS_SWCT_xxxxx].

The number of sampling points (Nx, Ny), the Offset, the shift and the distance values are represented in the following chapters by these logical views:

<insert figure with three representations>

These values are not defined inside SwRecordLayout(s) with fixed axis points.

Logical view:
The figure 3.27 illustrates the logical view of the SwRecordLayout FixIntMap. The SwRecordLayoutGroup with the shortLabel Val is shown in the lower part. Its elements are indexed by [AXIS 2, AXIS 1] from value (AXIS 2: = 1, AXIS 1: = 1) to value (AXIS 2: = -1, AXIS 1: = -1) there -1 gives the last value.
–Last change on issue 77620 comment 26–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.63 Specification Item TPS_SWCT_01749

Trace References:

none

Content:

The role SwAxisGeneric.SwAxisGeneric.swAxisType specifies the Identifiable.category of the fix axis according to TPS_SWCT_01748.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77620: Clarifiy description of FIX_AXIS

  Problem description:

  The AUTOSAR Meta Model is prepared for the description of FIX_AXIS but the SWC-T does not provide further information how to use it.

  Agreed solution:

  SWCT
  ====

  Create new chapter 5.4.4.1 Specification of fix axes

  In most cases the axes of a curve or map are accessible to a calibration software and it is possible to calibrate axes points and their corresponding values. There are cases, however, where axes are intentionally declared as fix and where no intention exists to change the properties of the axis ever[fn1].

  These axes are also known as fix axes.

  [TPS_SWCT_0xxx] Value of category for fix axis
  A fix axis shall be modeled as an SwCalprmAxis with attribute category set to the value FIX_AXIS.

  [TPS_SWCT_0xxx1] Sub-categories of fix axes
  There are different sub-categories of fix axes:
  * Fix axis where the distance between axis points can be computed according to a standardized algorithm. In this case, fix axes of arbitrary length can be described by feeding three arguments defined in the context of the axis description into the axis algorithm. Consequently, the memory footprint of different fix axis of this category is literally identical, independently of the number of axis points.
  The following variations exist:
  - Subcategory PAR, i.e. category = FIX_AXIS_PAR: the axis is created out of a starting value and a shift that creates further axis points as using a power-of-two algorithm. The details can be found in [27].
  - Subcategory PAR_DIST, i.e. category = FIX_AXIS_PAR_DIST: the axis is created

out of a starting value and an offset that adds further axis points with the distance given by offset. The details can be found in [27].
* Fix axis where the axis points are defined as a list of values directly in the axis definition. This variety boils down to
- Subcategory PAR_LIST, i.e. category = FIX_AXIS_PAR_LIST: the axis is created out of a list of numerical values that represent the axis points. The details can be found in [27].
These values of category shall be used for SwAxisType.

In other words, since the values that define a fix axis are modeled directly in the axis model there is no use case for assigning an initial value to a fix axis.

However, the calibration software may have still have access to axis points and values of the fix axis if these properties are specified in an A2L file. For this purpose McDataInstance needs to be set up properly. The details are explained in [7].

The modeling of of an axis of category FIX_AXIS_PAR is sketched in the following example model.

<insert attachment 4552>

The modeling of of an axis of category FIX_AXIS_PAR_DIST is sketched in the following example model.

<insert attachment 4553>

The modeling of of an axis of category FIX_AXIS_PAR_LIST is sketched in the following example model.

<insert attachment 4555>

Please note that the axis points and values of a fix axis are defined in the definition of the fix axis itself and therefore any initial value asigned to a fix axis would be ignored anyway. This might lead to confusion such that the initial value does not make it into the software. In order to avoid such confusion AUTOSAR does not support the definition of an initial value for a fix axis. This regulation is reflected in the existence of constr_xxx1.

[constr_0xxx1] No initialization for fix axis
an ApplicationValueSpecifciation taken to initialize an ApplicationPrimitiveDataType that contains a fix axis shall not contain initial values for the axis index of the fix axis

inside the ApplicationPrimitiveDataType.

Add table:

category of swAxisType category of SwGenericAxisParamType Multiplicity
FIX_AXIS_PAR OFFSET 1
SHIFT 1
FIX_AXIS_PAR_DIST OFFSET 1
DISTANCE 1
FIX_AXIS_PAR_LIST LIST 1

Add constraint that references the table.

Changes in the meta-model:

---

* Remove the mention of FIX_AXIS from the description of RuleBasedAxisCont.category (because fix axis are not initialized by means of a ValueSpecifiction)

* Remove the mention of FIX_AXIS from the description of SwAxisCont.category (because fix axis are not initialized by means of a ValueSpecifiction)

* make SwAxisGenericParam.vf ordered.

[fn1]: Typically, a calibration software does not have the ability to manipulate (or even inspect) the axis' properties by inspecting the ECU's memory.

[27]: ASAM MCD 2MC ASAP2 Interface Specification

[7]: Basic Software Module Description Template

GenBlprSupplement
=================

3.2.3 Record Layout: FixIntCur

This chapter describes the record layout for a curve with fixed axis points. Fixed axis exist in three categories: FIX_AXIS_PAR, FIX_AXIS_PAR_DIST and FIX_AXIS_PAR_LIST, see [TPS_SWCT_0xxx1].

The number of sampling points (Nx), the Offset, the shift and the distance values are represented in the following chapters by these logical views:

<insert figure with three representations>

These values are not defined inside SwRecordLayout(s) with fixed axis points.

Logical view:

The figure 3.9 illustrates the logical view of the SwRecordLayout FixIntCur. The SwRecordLayoutGroup with the shortLabel Val is shown in the lower part. Its elements are indexed by virtual [AXIS 1] which is fixed and of category FIX_AXIS_PAR and not defined inside this SwRecordLayout.

3.3.6 Record Layout: FixIntMap

This chapter describes the record layout for a map with fixed axis points. Fixed axis exist in three categories: FIX_AXIS_PAR, FIX_AXIS_PAR_DIST and FIX_AXIS_PAR_LIST, see [TPS_SWCT_xxxxx].

The number of sampling points (Nx, Ny), the Offset, the shift and the distance values are represented in the following chapters by these logical views:

<insert figure with three representations>

These values are not defined inside SwRecordLayout(s) with fixed axis points.

Logical view:
The figure 3.27 illustrates the logical view of the SwRecordLayout FixIntMap. The SwRecordLayoutGroup with the shortLabel Val is shown in the lower part. Its elements are indexed by [AXIS 2, AXIS 1] from value (AXIS 2: = 1, AXIS 1: = 1) to value (AXIS 2: = -1, AXIS 1: = -1) there -1 gives the last value.
–Last change on issue 77620 comment 26–

BW-C-Level:

Application	Specification	Bus
1	1	1

1.64 Specification Item TPS_SWCT_01750

Trace References:

none

Content:

The role SwAxisGeneric.SwAxisGeneric.swGenericAxisParam provides the actual numeric values for the definition of the axis.

The semantics of a provided numerical value is clarified by the attribute SwGenericAxisParamType.Identifiable.category where meta-class SwGenericAxisParamType is referenced in the role SwAxisType.swGenericAxisParamType.

RfCs affecting this spec item between releases 4.3.0 and 4.3.1:

- RfC #77620: Clarifiy description of FIX_AXIS

  Problem description:

  The AUTOSAR Meta Model is prepared for the description of FIX_AXIS but the SWC-T does not provide further information how to use it.

  Agreed solution:

  SWCT
  ====

  Create new chapter 5.4.4.1 Specification of fix axes

  In most cases the axes of a curve or map are accessible to a calibration software and it is possible to calibrate axes points and their corresponding values. There are cases, however, where axes are intentionally declared as fix and where no intention exists to change the properties of the axis ever[fn1].

  These axes are also known as fix axes.

  [TPS_SWCT_0xxx] Value of category for fix axis
  A fix axis shall be modeled as an SwCalprmAxis with attribute category set to the value FIX_AXIS.

  [TPS_SWCT_0xxx1] Sub-categories of fix axes
  There are different sub-categories of fix axes:
  * Fix axis where the distance between axis points can be computed according to a standardized algorithm. In this case, fix axes of arbitrary length can be described by feeding three arguments defined in the context of the axis description into the axis algorithm. Consequently, the memory footprint of different fix axis of this category is literally identical, independently of the number of axis points.
  The following variations exist:
  - Subcategory PAR, i.e. category = FIX_AXIS_PAR: the axis is created out of a starting value and a shift that creates further axis points as using a power-of-two

algorithm. The details can be found in [27].
- Subcategory PAR_DIST, i.e. category = FIX_AXIS_PAR_DIST: the axis is created out of a starting value and an offset that adds further axis points with the distance given by offset. The details can be found in [27].
* Fix axis where the axis points are defined as a list of values directly in the axis definition. This variety boils down to
- Subcategory PAR_LIST, i.e. category = FIX_AXIS_PAR_LIST: the axis is created out of a list of numerical values that represent the axis points. The details can be found in [27].
These values of category shall be used for SwAxisType.

In other words, since the values that define a fix axis are modeled directly in the axis model there is no use case for assigning an initial value to a fix axis.

However, the calibration software may have still have access to axis points and values of the fix axis if these properties are specified in an A2L file. For this purpose McDataInstance needs to be set up properly. The details are explained in [7].

The modeling of of an axis of category FIX_AXIS_PAR is sketched in the following example model.

<insert attachment 4552>

The modeling of of an axis of category FIX_AXIS_PAR_DIST is sketched in the following example model.

<insert attachment 4553>

The modeling of of an axis of category FIX_AXIS_PAR_LIST is sketched in the following example model.

<insert attachment 4555>

Please note that the axis points and values of a fix axis are defined in the definition of the fix axis itself and therefore any initial value asigned to a fix axis would be ignored anyway. This might lead to confusion such that the initial value does not make it into the software. In order to avoid such confusion AUTOSAR does not support the definition of an initial value for a fix axis. This regulation is reflected in the existence of constr_xxx1.

[constr_0xxx1] No initialization for fix axis an ApplicationValueSpecifciation taken to initialize an ApplicationPrimitiveDataType that contains a fix axis shall not contain initial values for the axis index of the fix axis inside the ApplicationPrimitiveDataType.

Add table:

category of swAxisType category of SwGenericAxisParamType Multiplicity
FIX_AXIS_PAR OFFSET 1
SHIFT 1
FIX_AXIS_PAR_DIST OFFSET 1
DISTANCE 1
FIX_AXIS_PAR_LIST LIST 1

Add constraint that references the table.

Changes in the meta-model:

---

* Remove the mention of FIX_AXIS from the description of RuleBasedAxisCont.category (because fix axis are not initialized by means of a ValueSpecifiction)

* Remove the mention of FIX_AXIS from the description of SwAxisCont.category (because fix axis are not initialized by means of a ValueSpecifiction)

* make SwAxisGenericParam.vf ordered.

[fn1]: Typically, a calibration software does not have the ability to manipulate (or even inspect) the axis' properties by inspecting the ECU's memory.

[27]: ASAM MCD 2MC ASAP2 Interface Specification

[7]: Basic Software Module Description Template

GenBlprSupplement
=================

3.2.3 Record Layout: FixIntCur

This chapter describes the record layout for a curve with fixed axis points. Fixed axis exist in three categories: FIX_AXIS_PAR, FIX_AXIS_PAR_DIST and FIX_AXIS_PAR_LIST, see [TPS_SWCT_0xxx1].

The number of sampling points (Nx), the Offset, the shift and the distance values are represented in the following chapters by these logical views:

<insert figure with three representations>

These values are not defined inside SwRecordLayout(s) with fixed axis points.

Logical view:

The figure 3.9 illustrates the logical view of the SwRecordLayout FixIntCur. The SwRecordLayoutGroup with the shortLabel Val is shown in the lower part. Its elements are indexed by virtual [AXIS 1] which is fixed and of category FIX_AXIS_PAR and not defined inside this SwRecordLayout.

3.3.6 Record Layout: FixIntMap

This chapter describes the record layout for a map with fixed axis points. Fixed axis exist in three categories: FIX_AXIS_PAR, FIX_AXIS_PAR_DIST and FIX_AXIS_PAR_LIST, see [TPS_SWCT_xxxxx].

The number of sampling points (Nx, Ny), the Offset, the shift and the distance values are represented in the following chapters by these logical views:

<insert figure with three representations>

These values are not defined inside SwRecordLayout(s) with fixed axis points.

Logical view:
The figure 3.27 illustrates the logical view of the SwRecordLayout FixIntMap. The SwRecordLayoutGroup with the shortLabel Val is shown in the lower part. Its elements are indexed by [AXIS 2, AXIS 1] from value (AXIS 2: = 1, AXIS 1: = 1) to value (AXIS 2: = -1, AXIS 1: = -1) there -1 gives the last value.
–Last change on issue 77620 comment 26–

BW-C-Level:

Application	**Specification**	**Bus**
1	1	1