

<b>Document Title</b>	Adaptive Platform Machine Configuration
<b>Document Owner</b>	AUTOSAR
<b>Document Responsibility</b>	AUTOSAR
<b>Document Identification No</b>	1112

<b>Document Status</b>	published
<b>Part of AUTOSAR Standard</b>	Adaptive Platform
<b>Part of Standard Release</b>	R24-11

<b>Document Change History</b>			
<b>Date</b>	<b>Release</b>	<b>Changed by</b>	<b>Description</b>
2024-11-27	R24-11	AUTOSAR Release Management	<ul style="list-style-type: none"> <li>• Initial release of the document</li> </ul>

## Disclaimer

This work (specification and/or software implementation) and the material contained in it, as released by AUTOSAR, is for the purpose of information only. AUTOSAR and the companies that have contributed to it shall not be liable for any use of the work.

The material contained in this work is protected by copyright and other types of intellectual property rights. The commercial exploitation of the material contained in this work requires a license to such intellectual property rights.

This work may be utilized or reproduced without any modification, in any form or by any means, for informational purposes only. For any other purpose, no part of the work may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The work has been developed for automotive applications only. It has neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

# Contents

1	Introduction	6
1.1	Document Conventions	6
2	Conceptual Aspects	8
2.1	Design and Target Configuration Models	8
2.1.1	Design Model	8
2.1.2	Target Configuration Models	9
2.2	Overview	9
2.3	Benefits of using an M1 model-based configuration approach	11
2.4	Modeling Strategies for deriving M1 Model Content	12
2.4.1	Only Design exists	12
2.4.2	Fully derive Design Model to Configuration Model	12
2.4.3	Partially derive Design Model to Configuration Model	13
2.4.4	Only Configuration exists	14
2.5	Vendor Specific Functional Cluster Definition	14
3	Overview	16
3.1	Abstract Structure	16
3.2	Definition Side	18
3.2.1	Definition Elements	19
3.2.2	Grouping of Configuration Elements	22
3.2.3	Emulation of Inheritance	24
3.2.4	Parameter	25
3.2.5	Reference	26
3.3	Value Side	27
3.3.1	Grouping of Configuration Values	30
3.3.2	Parameter	31
3.3.3	Reference	31
4	Details	33
4.1	Definition Side	33
4.1.1	Parameter	33
4.1.1.1	Enumeration Parameter	33
4.1.1.2	Boolean Parameter	35
4.1.1.3	Integer Parameter	35
4.1.1.4	Float Parameter	36
4.1.1.5	String Parameter	37
4.1.2	Reference	37
4.1.2.1	Reference to Container	38
4.1.2.2	Foreign Reference	40
4.1.2.3	Instance Reference	43
4.2	Value Side	46
4.2.1	Parameter	46
4.2.1.1	Numerical Parameter Value	47

4.2.1.2	Textual Parameter Value . . . . .	47
4.2.2	Reference . . . . .	48
A	Reference Material . . . . .	56
A.1	Terms and Abbreviations . . . . .	56
A.2	Imposition Times of Constraints . . . . .	56
B	Examples . . . . .	58
C	History of Constraints and Specification Items . . . . .	59
C.1	Constraint and Specification Item History of this Document according to AUTOSAR R24-11 . . . . .	59
C.1.1	Added Specification Items in R24-11 . . . . .	59
C.1.2	Changed Specification Items in R24-11 . . . . .	60
C.1.3	Deleted Specification Items in R24-11 . . . . .	60
C.1.4	Added Constraints in R24-11 . . . . .	61
C.1.5	Changed Constraints in R24-11 . . . . .	61
C.1.6	Deleted Constraints in R24-11 . . . . .	61
D	Mentioned Class Tables . . . . .	62

## References

- [1] Specification of ECU Configuration  
AUTOSAR\_CP\_TPS\_ECUConfiguration
- [2] Meta Model  
AUTOSAR\_FO\_MMOD\_MetaModel
- [3] Meta Model-generated XML Schema  
AUTOSAR\_FO\_MMOD\_XMLSchema
- [4] Glossary  
AUTOSAR\_FO\_TR\_Glossary

# 1 Introduction

This document describes the structure of the configuration (M2-level) model applicable to the creation of *target-configuration* (as opposed to *design-level*) content on the *AUTOSAR adaptive platform*.

The M2 model described in this document provides the structural basis for the creation of target-configuration (M1) models for individual functional clusters.

## 1.1 Document Conventions

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

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

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

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

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

Please note that constraints are not supposed to be enforceable at any given time in an AUTOSAR workflow. During the development of a model, constraints may legitimately be violated because an incomplete model will obviously show inconsistencies.

However, at specific points in the workflow, constraints shall be enforced as a safeguard against misconfiguration.

The points in the workflow where constraints shall be enforced, sometimes also known as the "binding time" of the constraint, are different for each model category, e.g. on the classic platform, the constraints defined for software-components are typically enforced prior to the generation of the RTE while the constraints against the definition of an Ecu extract shall be applied when the Ecu configuration for the Com stack is created.

For each document, possible binding times of constraints are defined and the binding times are typically mentioned in the constraint themselves to give a proper orientation for implementers of AUTOSAR authoring tools.

Let **AUTOSAR** be an example of a typical class table. The first rows in the table have the following meaning:

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

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

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

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

The headers in the table have the following meaning:

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

**Type:** The type of an attribute of the class.

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

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

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

Please note that the chapters that start with a letter instead of a numerical value represent the appendix of the document. The purpose of the appendix is to support the explanation of certain aspects of the document and does not represent binding conventions of the standard.

## 2 Conceptual Aspects

### 2.1 Design and Target Configuration Models

If you look at AUTOSAR Models from a high vantage point, two kinds of models can be identified: design models and `target-configuration` models.

The purpose of this section is to give a rough intuition about the distinction between the two model flavors. It does not cover edge cases and messy details. While on first glance, this distinction is relatively obvious, there are also model elements that combine aspects of design and configuration. When looking at the details, the boundary is often blurry.

Please note that in the past, the term “deployment” was often used to refer to some aspects of “`target-configuration`”. “Deployment”, however, is heavily overloaded with many meanings.

To avoid confusion, it was decided to introduce the term “target-configuration”. It refers to configuration whose scope is local to a target, like `ECU Configuration (EcuC)` for an `EcuInstance` on the *AUTOSAR classic Platform*, and `Adaptive Platform Machine Configuration (APMC)` for a `Machine` on the *AUTOSAR adaptive Platform*.

#### 2.1.1 Design Model

A rough criterion for identifying a model element as design-related, is whether or not the creation of the model element would typically be done in a System Designer role (usually related to the OEM scope).

Related to this point of view, is the question whether the model element contributes to making the overall system (beyond the scope of a single compute node) work.

In other words, a design-related model describes the external behavior, as in “the DEXT describes the external behavior of the `Dcm/Dem/Fim` (on the *AUTOSAR classic Platform*) or `Dm` (on the *AUTOSAR adaptive Platform*)”.

Examples for this perspective would be: the baud-rate of a network, or the definition of a CAN Id, or an application interface that acts as a contract between the application layer and the functional clusters.

As a consequence, many design-related model elements can't be changed during the `target-configuration` phase, or else the system will break.

Another distinguishing characteristic of a design-related model element is that it is used to exchange information between different roles. For example between System Designer and Integrator, or between Application Developer and System Designer.



### 2.1.2 Target Configuration Models

The `target-configuration` model contains model elements whose scope is mostly local to a given compute node. Therefore, these elements are usually created during integration.

Some `target-configuration`-level model elements maintain a dedicated relation to design-level model elements, such that the `target-configuration` can only be considered complete, if the information from the design perspective is considered for the `target-configuration`.

There are also `target-configuration` elements that are directly derived from the content of a design-level model element. This strategy is heavily used on the *AUTOSAR classic Platform*.

It is not yet clarified whether it will also be relevant on the *AUTOSAR adaptive Platform*.

In contrast to the *AUTOSAR classic Platform* ECU Configuration model (`ECuC`), which often aims for a standalone model containing the complete information required to configure the BSW Stack, the `APMC` assumes that parts of the design model will be available on the target (usually not in ARXML format, but converted into a machine specific one).

For example, if a Service Interface can be used directly to configure an Adaptive Service, then there is no need to create a copy of that information in `APMC`.

## 2.2 Overview

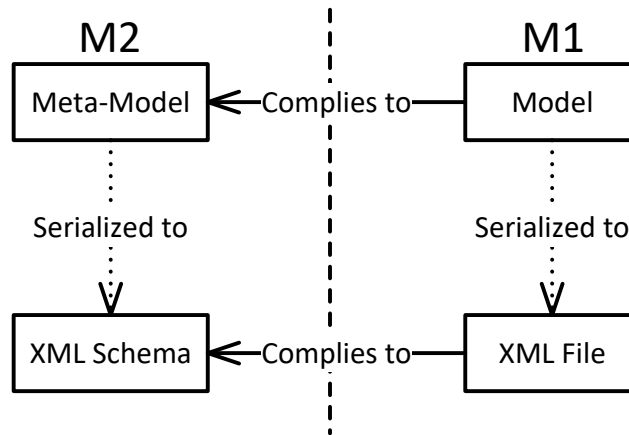
The big picture of the `target-configuration` configuration on the *AUTOSAR adaptive Platform* looks very similar to the configuration on the *AUTOSAR classic Platform*, as described by the TPS ECU Configuration [1].

The difference is that the approach on the *AUTOSAR adaptive Platform* is simpler because it is not constrained by some of the boundary conditions that apply for the *AUTOSAR classic Platform*.

For example, `APMC` does not have to provide support for code generation (e.g. symbolic names), because this is currently not standardized on `AP`. In addition, the application of variation (let alone post-build variation) is not yet clarified on `AP`.

From high above, the configuration model differs from the design model such that the design model is build based on the AUTOSAR meta-model [2] (also known as the M2 model), whereas the configuration model follows the specification of the configuration parameters that in turn is also based on the AUTOSAR meta-model, but one meta-level below (also known as the M1 model).

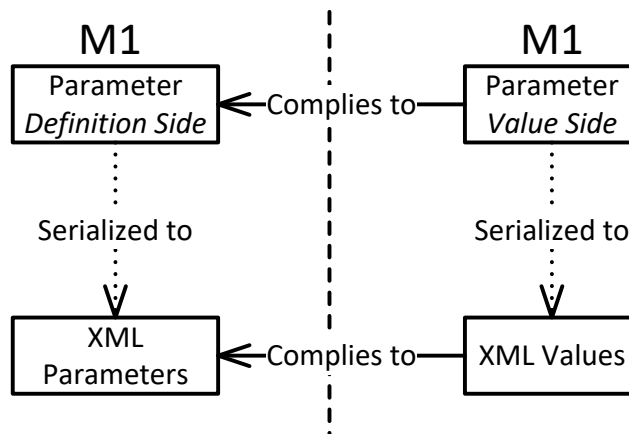
As depicted by Figure 2.1, the difference between the M2 and the M1 model is that the M2 model is the source for the definition of (and serialization to) the XML schema [3]. On the other hand, an M1 model is serialized to an ARXML file.



**Figure 2.1: Visualization of different meta-levels from the serialization perspective**

As also depicted in Figure 2.1, the M2 model has the job to describe the “grammar” in which the M1 model is formulated. An M1 model is expected to follow the structure defined by the corresponding M2 model.

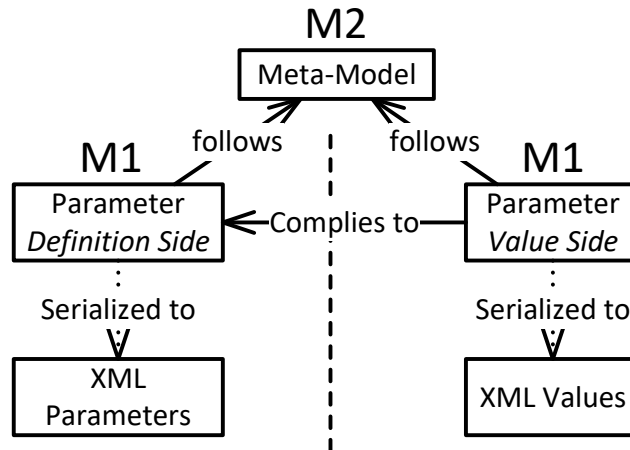
On the serialization level, this means that the serialization of an M1 as an ARXML file shall validate against the serialization of the M2 model in the form of the AUTOSAR XML schema.



**Figure 2.2: Visualization of the difference between parameter definition and parameter value**

It is also possible to establish a relation between two models such that the first one establishes the “grammar” for the second model. This relation is depicted in Figure 2.2. On the file level, this means that the “value side” file shall follow the structure defined by the “definition side” file.

To summarize, the “value side” and the “definition side” on the M1 level have to follow the **structure** defined by the M2 model. In addition, the “value side” has to follow the **structure** of “definition side”. This aspect is depicted in Figure 2.3.



**Figure 2.3: Visualization of the overall relation between meta-levels and “definition” vs. “value”**

## 2.3 Benefits of using an M1 model-based configuration approach

In comparison to an equivalent implementation of the configuration model for the *AUTOSAR adaptive Platform* on the M2 level, the chosen approach comes with the following benefits:

- Much simpler model structure (easier to translate into vendor-specific *target-configuration* representations)
- Ability to define default values for specific configuration parameters
- Ability to create the model using the function-cluster-specific terminology
- Ability to define specific multiplicities (other than being restricted to 0..1 and 0..\*)
- Ability to respond to changing requirements without having to wait for a schema change
- Ability to define vendor-specific extensions as first-class citizen
- Ability to respond to changing requirements without consideration of backwards compatibility on schema level
- Clear separation between the design-workflow and configuration-workflow

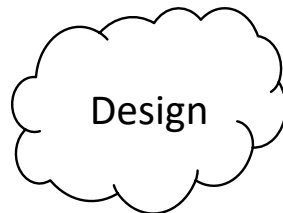
## 2.4 Modeling Strategies for deriving M1 Model Content

The configuration of the *AUTOSAR adaptive Platform* is not a stand-alone approach. Configuration models in many<sup>1</sup> cases need to refer to model elements from the design model in order to provide the complete picture.

In principle, different modeling strategies for combining the information from design model elements with the information on configuration level could be conceived.

### 2.4.1 Only Design exists

In some rare cases, it is possible to take the configuration of a functional cluster entirely from the design model, i.e. no configuration model is required in the first place.



**Figure 2.4: Visualization of the modeling strategy in which only a design model exists**

### 2.4.2 Fully derive Design Model to Configuration Model

This modeling strategy is, to some extent, similar to the one described in section 2.4.4, as its goal is to derive all relevant information from the design model into a respective representation in the configuration model.

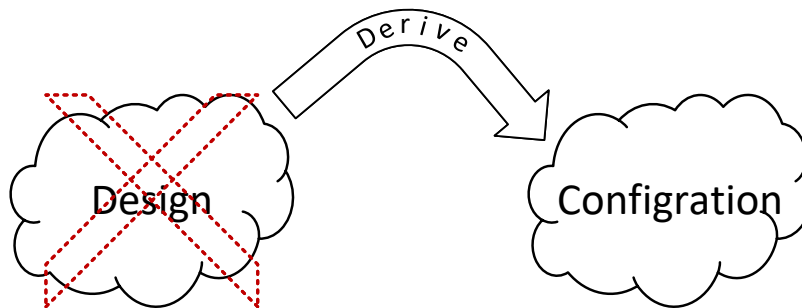
After this derivation process completes, all relevant design information has become part of the configuration model. This means that the design model is no longer needed and technically might just as well be deleted.

This modeling strategy is very similar to the approach taken on the *AUTOSAR classic Platform* for the derivation of the configuration of the Com stack. While the derivation of the Com stack works very well, it has also been criticized for the amount of information duplication.

In other words, the configuration model tends to become big and complex if this modeling strategy is applied.

---

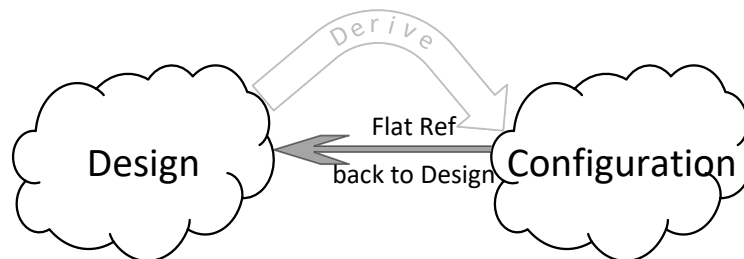
<sup>1</sup>but most likely not all cases



**Figure 2.5: Visualization of the modeling strategy in which the design model is fully derived to the configuration**

### 2.4.3 Partially derive Design Model to Configuration Model

If the full derivation of the design model to the configuration model, as described in section 2.4.2 is considered to produce too complex models, a possible alternative could be to just derive the design model partially<sup>2</sup> and establish references to the still existing design model wherever these references can be modeled as flat references.



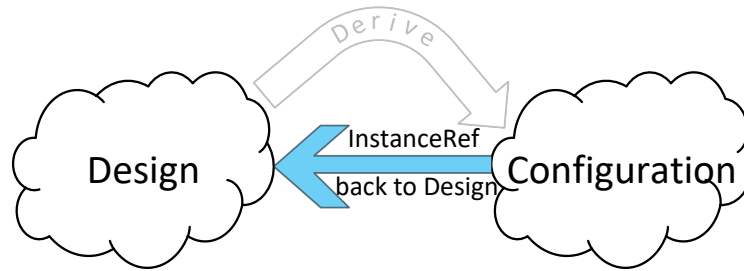
**Figure 2.6: Visualization of the modeling strategy in which the configuration model is partially derived out of the design model and maintains flat references into the design model**

Consequently, the part of the design model that would require the creation of flat references does not make use of the AUTOSAR type-prototype pattern.

The alternative to using flat references would be the usage of instanceRefs. Again, the derivation of the configuration model from the design model is optional.

The core characteristics of this modeling strategy is the utilization of instanceRefs, which also means that this modeling strategy is targeting the part of the meta-model where the type-prototype pattern has been used to create the meta-model.

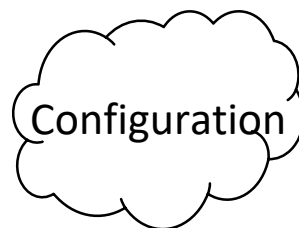
<sup>2</sup>In the extreme case: not at all, but this is a question of principle.



**Figure 2.7: Visualization of the modeling strategy in which the configuration model is partially derived out of the design model and maintains instanceRefs into the design model**

### 2.4.4 Only Configuration exists

This modeling strategy is based on the idea that there is no relevant design model, and all the configuration is exclusively done on the M1 level. Such a case is certainly special and does not represent a mainstream possibility.



**Figure 2.8: Visualization of the modeling strategy in which only a configuration model exists**

## 2.5 Vendor Specific Functional Cluster Definition

The Functional Cluster Definition that is standardized by AUTOSAR is the base for the configuration of the Functional Cluster. Out of the Standardized Functional Cluster Definition the Vendor-Specific Functional Cluster Definition can be derived.

Such a Vendor-Specific Functional Cluster Definition is allowed to define additional vendor-specific extensions (e.g. additional containers, parameters or references) with the same set of language elements as the Standardized Functional Cluster Definition.

This eases the development of tools and introduces the possibility to standardize vendor-specific parameters at a later point in time in AUTOSAR.

**[AP\_TPS\_APMC\_01038] `ApmcFunctionalClusterDef` categories**

*Status:* DRAFT

[The `category` attribute of `ApmcFunctionalClusterDef` defines the kind of the Functional Cluster Definition:

- `STANDARDIZED_CLUSTER_DEFINITION` - Functional Cluster Definition that is standardized and released by AUTOSAR
- `VENDOR_SPECIFIC_CLUSTER_DEFINITION` - Vendor Specific Cluster Definition that refines a `STANDARDIZED_CLUSTER_DEFINITION` with vendor specific extensions

]

**[constr\_9322] `ApmcFunctionalClusterDef` category restriction**

*Status:* DRAFT

*Imposition time:* IT\_CfgFc

[The following values for the attribute `category` of `ApmcFunctionalClusterDef` are reserved by the AUTOSAR Standard:

- `STANDARDIZED_CLUSTER_DEFINITION`
- `VENDOR_SPECIFIC_CLUSTER_DEFINITION`

]

Please note that custom values for attribute `category` attribute of `ApmcFunctionalClusterDef` are currently not supported.

### 3 Overview

This chapter provides a description of the configuration elements with a focus on the overview and general modeling strategy.

As already hinted, the modeling approach for the `target-configuration` is split into two parts:

- The “definition side” is used to define the structure of the configuration for specific parts of the `target-configuration`, e.g. the configuration of a specific functional cluster.
- The “value side” is used to create values for specific configuration items according to the structure defined in the “definition side”.

#### 3.1 Abstract Structure

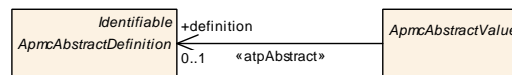
##### [AP\_TPS\_APMC\_01000] Abstract relation between the “value side” and the “definition side”

*Status:* DRAFT

[The relation between the “value side” and the “definition side” is that the “value side” needs to follow the structure defined by the “definition side”.

The concrete relation between a value and the corresponding definition is defined by a reference from the “value side” model element to the “definition side” model element.

This reference is established on an abstract level by the reference in the role `ApmcAbstractValue.definition` to `ApmcAbstractDefinition`. This reference is inherited to the respective corresponding sub-classes of `ApmcAbstractValue` and `ApmcAbstractDefinition`.]



**Figure 3.1: Abstract structure of the `target-configuration` configuration**

Please note that there is no 1:1 relation between an `ApmcAbstractValue` and an `ApmcAbstractDefinition`. It is possible to create a collection with an upper multiplicity greater than 1 within the `APMC` model and therefore the same `ApmcAbstractDefinition` may be referenced by more than one `ApmcAbstractValue`.



<b>Class</b>	<b>ApmcAbstractDefinition</b> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::AbstractStructure			
<b>Note</b>	This abstract meta-class implements the "definition" side of the abstract structure for the machine configuration. <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	ARObject, <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Subclasses</b>	<a href="#">ApmcContainerDef</a> , <a href="#">ApmcContainerElementDef</a> , <a href="#">ApmcFunctionalClusterDef</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 3.1: ApmcAbstractDefinition**

<b>Class</b>	<b>ApmcAbstractValue</b> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::AbstractStructure			
<b>Note</b>	This abstract meta-class implements the "value" side of the abstract structure for the machine configuration. <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	ARObject			
<b>Subclasses</b>	<a href="#">ApmcContainerElementValue</a> , <a href="#">ApmcContainerValue</a> , <a href="#">ApmcFunctionalClusterValue</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
definition	<a href="#">ApmcAbstractDefinition</a>	0..1	ref	This reference implements the relation of "value" side to "definition" side in the abstract structure of the machine configuration model. <b>Stereotypes:</b> atpAbstract <b>Tags:</b> atp.Status=candidate

**Table 3.2: ApmcAbstractValue**

### [constr\_10579] Existence of [definition](#)

*Status:* DRAFT

*Imposition time:* IT\_CfgFc

[In the context of an [ApmcAbstractValue](#), the reference in the role [definition](#) shall exist.]

As a rough categorization, the following configuration elements are provided:

- Functional cluster
- Container
- Parameter
- Reference

An overview of the modeling on the “definition side” is given in section [3.2](#). On the same level of detail, the modeling on the “value side” is described in section [3.3](#).

## 3.2 Definition Side

### [AP\_TPS\_APMC\_01001] Root element on the “definition side”

Status: DRAFT

[Meta-class [ApmcFunctionalClusterDef](#) represents the root element of the `target-configuration` configuration on the “definition side”.]

### [AP\_TPS\_APMC\_01002] Semantics of [ApmcDefinitionCollection](#)

Status: DRAFT

[Meta-class [ApmcDefinitionCollection](#) acts (as the name already suggests) as a collector, such that model elements that need to refer to a number of different [ApmcFunctionalClusterDefs](#) can instead simply create one reference to the [ApmcDefinitionCollection](#) that in turn refers to the affected [ApmcFunctionalClusterDefs](#) in the role `functionalCluster`.]

<b>Class</b>	<b>ApmcDefinitionCollection</b>			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::DefinitionSide			
<b>Note</b>	This class represents the entry point of the target-configuration on the "definition" side. <b>Tags:</b> atp.Status=candidate atp.recommendedPackage=SoftwareClusterDefinitions			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">AtpBlueprint</a> , <a href="#">AtpBlueprintable</a> , <a href="#">CollectableElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a> , <a href="#">UploadablePackageElement</a>			
<b>Aggregated by</b>	ARPackage.element			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
functional Cluster	<a href="#">ApmcFunctionalClusterDef</a>	*	ref	This reference identifies the collection of functional clusters that are configured (partially) in the enclosing <a href="#">AmpcSoftwareClusterDefinition</a> . <b>Tags:</b> atp.Status=candidate

**Table 3.3: ApmcDefinitionCollection**

<b>Class</b>	<b>ApmcFunctionalClusterDef</b>			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::DefinitionSide			
<b>Note</b>	This class represents the entry point into the configuration of a single functional cluster on the AUTOSAR adaptive platform on the "definition" side. <b>Tags:</b> atp.Status=candidate atp.recommendedPackage=FunctionalClusterDefinitions			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">ApmcAbstractDefinition</a> , <a href="#">ApmcDefinitionElement</a> , <a href="#">AtpBlueprint</a> , <a href="#">AtpBlueprintable</a> , <a href="#">CollectableElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a> , <a href="#">UploadablePackageElement</a>			
<b>Aggregated by</b>	ARPackage.element			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>





Class	ApmcFunctionalClusterDef			
container	<a href="#">ApmcContainerDef</a>	*	aggr	This aggregation represents the collection of containers aggregated directly by the functional cluster configuration definition.  <b>Stereotypes:</b> atpSplitable <b>Tags:</b> atp.Splitkey=container.shortName atp.Status=candidate

**Table 3.4: ApmcFunctionalClusterDef**

### 3.2.1 Definition Elements

Meta-class [ApmcDefinitionElement](#) is used to inherit some essential and commonly used attributes to the various classes on the “definition side” of the [target-configuration](#).

Class	ApmcDefinitionElement (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::DefinitionSide			
<b>Note</b>	This abstract class contributes basic properties to the modeling of classes on the "definition" side of the target-configuration.  <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	<i>ARObject</i>			
<b>Subclasses</b>	<a href="#">ApmcContainerDef</a> , <a href="#">ApmcContainerElementDef</a> , <a href="#">ApmcFunctionalClusterDef</a>			
Attribute	Type	Mult.	Kind	Note
element Creation	<a href="#">ApmcElementCreation Enum</a>	0..1	attr	This attribute gives a hint about how (in terms of the AUTOSAR workflow) the model element comes into existence.  <b>Tags:</b> atp.Status=candidate
lowerMultiplicity	PositiveInteger	0..1	attr	This attribute specifies the lower multiplicity of the respective model element.  <b>Tags:</b> atp.Status=candidate
upperMultiplicity	PositiveInteger	0..1	attr	This attribute specifies the upper multiplicity of the respective model element.  <b>Tags:</b> atp.Status=candidate
upperMultiplicity IsInfinite	Boolean	0..1	attr	This attribute indicates (if set to true) that the upper multiplicity is set to "infinite".  <b>Tags:</b> atp.Status=candidate

**Table 3.5: ApmcDefinitionElement**

### [AP\_TPS\_APMC\_01003] Specification of element multiplicity on the “definition side”

*Status:* DRAFT

[Contrary to the situation on the M2 model, multiplicities can't be defined on an association, but on the associated element itself. For this purpose, the attributes

- [ApmcDefinitionElement.lowerMultiplicity](#)

- `ApmcDefinitionElement.upperMultiplicity`
- `ApmcDefinitionElement.upperMultiplicityIsInfinite`

are provided.

The purpose of attribute `upperMultiplicityIsInfinite` is to indicate that the upper multiplicity is not defined by the model element itself, but depends on the usage of the model element.

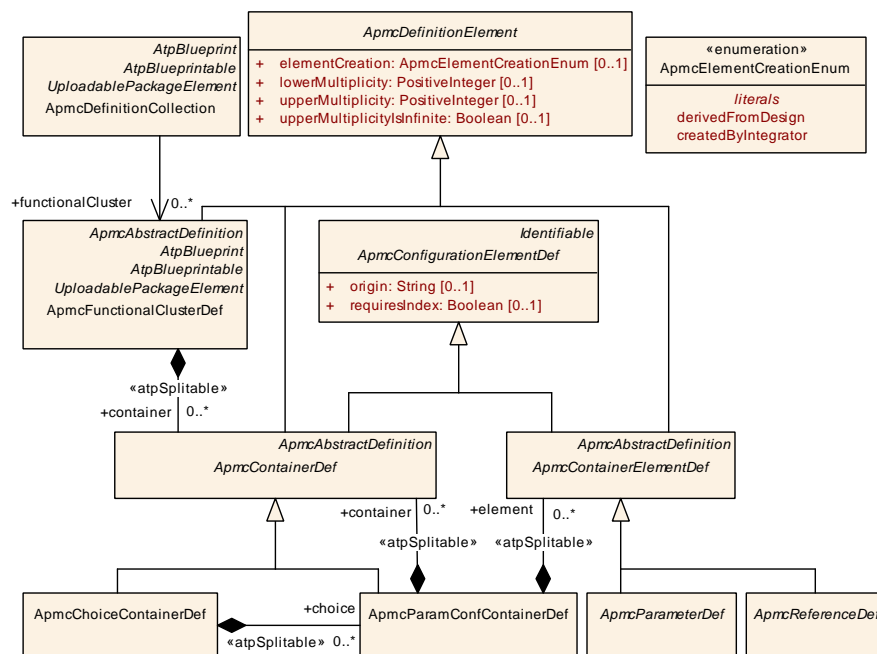
This intention cannot be communicated by the attribute `upperMultiplicity` because the latter expects a concrete value.]

Please note that the definition of multiplicities on the `target-configuration` is not subject to restrictions in terms of strict modeling patterns<sup>1</sup>. Any constellations are supported, as long as the value of `lowerMultiplicity` is smaller or equal to the value `ApmcDefinitionElement.upperMultiplicity` if this attribute exists.

**[constr\_10580] Restriction regarding the value of `ApmcDefinitionElement.lowerMultiplicity` vs. `ApmcDefinitionElement.lowerMultiplicity`**

Status: DRAFT  
Imposition time: IT\_CfgFc

[In the context of any given `ApmcDefinitionElement`, if attribute `upperMultiplicity` exists, then its value shall be greater or equal to the value of the attribute `lowerMultiplicity` in the same context.]



**Figure 3.2: Overview of the `target-configuration` definition**

<sup>1</sup>like on the M2 model

**[AP\_TPS\_APMC\_01004] Semantics of attribute `ApmcDefinitionElement.elementCreation`**

Status: DRAFT

[Attribute `ApmcDefinitionElement.elementCreation` can be used to indicate the intended creation of the respective model element on the “definition side”. The attribute allows for specifying the intention that either

- the model element is foreseen to be derived from an equivalent model element in the design domain (in which case the value `ApmcElementCreationEnum.derivedFromDesign` applies) or
- the model element is intended to be created by an integrator (in which case the value `ApmcElementCreationEnum.createdByIntegrator` applies).

This information may be helpful for designers of tool workflow such that the handling of the parameters in the configuration workflow can be properly considered.]

<b>Enumeration</b>	<b>ApmcElementCreationEnum</b>
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::DefinitionSide
<b>Note</b>	This enumeration defines different approaches for the creation of a model element in the context of the target-configuration. <b>Tags:</b> atp.Status=candidate
<b>Aggregated by</b>	<code>ApmcDefinitionElement.elementCreation</code>
<b>Literal</b>	<b>Description</b>
createdByIntegrator	This literal indicates that the model element is explicitly created by an integrator during the deployment phase of the AUTOSAR workflow. <b>Tags:</b> atp.EnumerationLiteralIndex=1 atp.Status=candidate
derivedFromDesign	This literal indicates that the model element is derived from information available in the design step of the AUTOSAR workflow. <b>Tags:</b> atp.EnumerationLiteralIndex=0 atp.Status=candidate

**Table 3.6: ApmcElementCreationEnum**

**[AP\_TPS\_APMC\_01005] Semantics of attribute `ApmcConfigurationElementDef.origin`**

Status: DRAFT

[Attribute `ApmcConfigurationElementDef.origin` shall be used to indicate whether the respective model element has been defined as either

- a part of the AUTOSAR standard (in which case the value of the attribute shall be `AUTOSAR_APMC`) or
- as a vendor-specific configuration element (in which case the value of the attribute shall be clearly marked as vendor-specific by using the vendor name in the value of the attribute).

]

<b>Class</b>	<b>ApmcConfigurationElementDef</b> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::DefinitionSide			
<b>Note</b>	This abstract class provides some attributes that are relevant as a contribution to the definition of meta-classes in the context the "definition" side of the target-configuration. <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	ARObject, <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Subclasses</b>	<a href="#">ApmcContainerDef</a> , <a href="#">ApmcContainerElementDef</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
origin	String	0..1	attr	This attribute provides the information whether the definition of the configuration element is standardized by AUTOSAR or whether it represents a vendor-specific configuration element. <b>Tags:</b> atp.Status=candidate
requiresIndex	Boolean	0..1	attr	This attribute indicates whether the elements of a collection of the enclosing model element carry an index value to allow for the identification of order.  In other words, the order is not established on the syntactical level by the appearance in the ARXML files but on a semantic level by evaluating the value of the index attribute. <b>Tags:</b> atp.Status=candidate

**Table 3.7: ApmcConfigurationElementDef**

In some cases, it is necessary to create collections where the order of elements in the collection is significant. On the M2 level, this can be achieved by using the qualifier “ordered” at the definition of the collection.

This ability does not exist on the M1 level and therefore the ordering is supported by the explicit definition of an index value for the elements of a collection.

### [AP\_TPS\_APMC\_01013] Semantics of attribute [ApmcConfigurationElementDef.requiresIndex](#)

*Status:* DRAFT

[Attribute [ApmcConfigurationElementDef.requiresIndex](#) has been defined to allow for the creation of *ordered collections*. The position of an element of the collection within the collection is defined by an index value on the “value side”.]

## 3.2.2 Grouping of Configuration Elements

The container level within the configuration approach exists for several purposes (see also Figure 3.2):

- Provide a simple grouping mechanism for other model elements, see [ApmcContainerElementDef](#), resp. the aggregation in the role [ApmcParamConfContainerDef.element](#).
- Provide a way to create nested structures of such a grouping, see [ApmcParamConfContainerDef.container](#).
- Provide the ability to become a target for references that create relation between containers on different nesting levels, see section 4.1.
- Emulate “inheritance” by means of choice-containers, see [ApmcChoiceContainerDef](#).

### [AP\_TPS\_APMC\_01006] Grouping of configuration items on the level of a container

Status: DRAFT

[The grouping of configuration items is implemented by means of the roles:

- [ApmcParamConfContainerDef.element](#) for the aggregation of “primitive” configuration items.
- [ApmcParamConfContainerDef.container](#) for the creation of nested levels of containers.

]

<b>Class</b>	<b>ApmcContainerDef</b> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::DefinitionSide			
<b>Note</b>	This abstract meta-class represents a container on the "definition" side of the target-configuration. <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	<a href="#">ARObject</a> , <a href="#">ApmcAbstractDefinition</a> , <a href="#">ApmcConfigurationElementDef</a> , <a href="#">ApmcDefinitionElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Subclasses</b>	<a href="#">ApmcChoiceContainerDef</a> , <a href="#">ApmcParamConfContainerDef</a>			
<b>Aggregated by</b>	<a href="#">ApmcFunctionalClusterDef.container</a> , <a href="#">ApmcParamConfContainerDef.container</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 3.8: ApmcContainerDef**

<b>Class</b>	<b>ApmcContainerElementDef</b> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::DefinitionSide			
<b>Note</b>	This abstract meta-class represents an element of a container on the "definition" side of the target-configuration. <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	<a href="#">ARObject</a> , <a href="#">ApmcAbstractDefinition</a> , <a href="#">ApmcConfigurationElementDef</a> , <a href="#">ApmcDefinitionElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Subclasses</b>	<a href="#">ApmcParameterDef</a> , <a href="#">ApmcReferenceDef</a>			





<b>Class</b>	<b>ApmcContainerElementDef</b> (abstract)			
<b>Aggregated by</b>	<a href="#">ApmcParamConfContainerDef.element</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 3.9: ApmcContainerElementDef**

<b>Class</b>	<b>ApmcParamConfContainerDef</b>			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::DefinitionSide			
<b>Note</b>	This meta-class represents the definition of container (on the "definition" side of the target-configuration) that carries the definition of parameters and references. <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	<a href="#">ARObject</a> , <a href="#">ApmcAbstractDefinition</a> , <a href="#">ApmcConfigurationElementDef</a> , <a href="#">ApmcContainerDef</a> , <a href="#">ApmcDefinitionElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Aggregated by</b>	<a href="#">ApmcChoiceContainerDef.choice</a> , <a href="#">ApmcFunctionalClusterDef.container</a> , <a href="#">ApmcParamConfContainerDef.container</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
container	<a href="#">ApmcContainerDef</a>	*	aggr	This aggregation represents the collection of sub-containers of the enclosing container. By using this aggregation arbitrary nested structures of containers can be created. <b>Stereotypes:</b> atpSplitable <b>Tags:</b> atp.Splitkey=container.shortName atp.Status=candidate
element	<a href="#">ApmcContainerElementDef</a>	*	aggr	This aggregation represents the collection of elements in the container. <b>Stereotypes:</b> atpSplitable <b>Tags:</b> atp.Splitkey=element.shortName atp.Status=candidate

**Table 3.10: ApmcParamConfContainerDef**

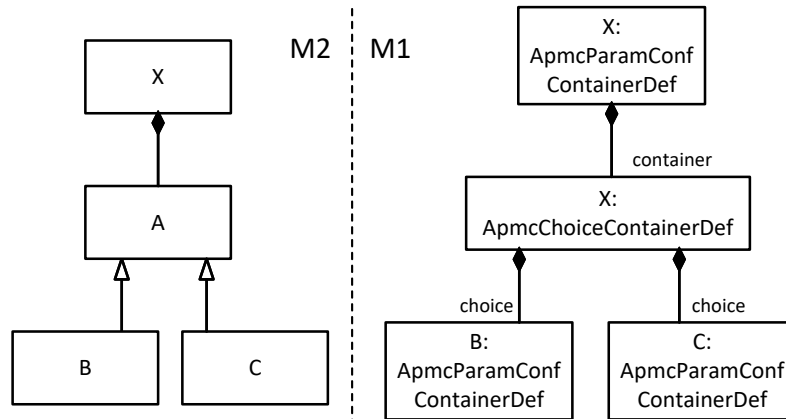
### 3.2.3 Emulation of Inheritance

#### [AP\_TPS\_APMC\_01007] Semantics of a choice container

*Status:* DRAFT

[The existence of an [ApmcChoiceContainerDef](#) implements a structural alternative in the `target-configuration`. The behavior of the [ApmcChoiceContainerDef](#) has a similar effect as the creation of sub-classes on the M2 model.]





**Figure 3.3: Visualization of the semantics of a choice container**

The statement made by [AP\_TPS\_APMC\_01006] is visualized in Figure 3.3, in which the equivalent of an inheritance relation in the M2 model (left part of Figure 3.3) is created by means of the `target-configuration` M1 model (right part of Figure 3.3).

<b>Class</b>	<b>ApmcChoiceContainerDef</b>			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::DefinitionSide			
<b>Note</b>	This meta-class represents the ability to specify a choice-semantics on the container level (on the "definition" side of the target-configuration).  The choice semantics can be utilized to emulate inheritance in the M1 model.  <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	<a href="#">ARObject</a> , <a href="#">ApmcAbstractDefinition</a> , <a href="#">ApmcConfigurationElementDef</a> , <a href="#">ApmcContainerDef</a> , <a href="#">ApmcDefinitionElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Aggregated by</b>	<a href="#">ApmcFunctionalClusterDef.container</a> , <a href="#">ApmcParamConfContainerDef.container</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
choice	<a href="#">ApmcParamConf ContainerDef</a>	*	aggr	This collection of containers represents the content of the enclosing choice container.  <b>Stereotypes:</b> atpSplitable <b>Tags:</b> atp.Splitkey=choice.shortName atp.Status=candidate

**Table 3.11: ApmcChoiceContainerDef**

### 3.2.4 Parameter

#### [AP\_TPS\_APMC\_01008] Definition of configuration parameter

*Status:* DRAFT

[A configuration parameter is used to adjust a specific atomic aspect of the `target-configuration`. Configuration parameters are available in different flavors, depending on the data type of the information they need to carry.

For this purpose, different data-type-specific subclasses of the abstract class [ApmcParameterDef](#) exist on the “definition side” to describe the definition of a parameter of a certain nature.

A Configuration parameter can only be defined in the context of a [ApmcParamConfContainerDef](#).]

The details are explained in section [4.1.1](#).

The abstract meta-class [ApmcParameterDef](#) acts as a base class for the specific kinds of parameter definition.

<b>Class</b>	<a href="#">ApmcParameterDef</a> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::DefinitionSide			
<b>Note</b>	This meta-class represents a parameter on the "definition" side of the adaptive platform. The concrete nature of the parameter is clarified by a subclass. <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	<a href="#">ARObject</a> , <a href="#">ApmcAbstractDefinition</a> , <a href="#">ApmcConfigurationElementDef</a> , <a href="#">ApmcContainerElementDef</a> , <a href="#">ApmcDefinitionElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Subclasses</b>	<a href="#">ApmcBooleanParamDef</a> , <a href="#">ApmcEnumerationParamDef</a> , <a href="#">ApmcFloatParamDef</a> , <a href="#">ApmcIntegerParamDef</a> , <a href="#">ApmcStringParamDef</a>			
<b>Aggregated by</b>	<a href="#">ApmcParamConfContainerDef</a> .element			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 3.12: ApmcParameterDef**

### 3.2.5 Reference

#### [AP\_TPS\_APMC\_01009] Definition of reference

*Status:* DRAFT

[In the context of the [target-configuration](#), references can be used to either create a semantical link between different parts of the [target-configuration](#) or to associate a part of the [target-configuration](#) with an element from the design model.]

The details are explained in section [4.1.2](#).

The abstract meta-class [ApmcReferenceDef](#) acts as a base class for the specific kinds of reference definition.

<b>Class</b>	<i>ApmcReferenceDef</i> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::DefinitionSide			
<b>Note</b>	This meta-class represents a reference on the "definition" side of the adaptive platform. The concrete nature of the reference is clarified by a subclass. <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	<i>ARObject</i> , <i>ApmcAbstractDefinition</i> , <i>ApmcConfigurationElementDef</i> , <i>ApmcContainerElementDef</i> , <i>ApmcDefinitionElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
<b>Subclasses</b>	<i>ApmcAbstractForeignReferenceDef</i> , <i>ApmcAbstractInstanceReferenceDef</i> , <i>ApmcChoiceContainerReferenceDef</i> , <i>ApmcContainerReferenceDef</i>			
<b>Aggregated by</b>	<i>ApmcParamConfContainerDef</i> .element			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 3.13: ApmcReferenceDef**

### 3.3 Value Side

The value side of the `target-configuration` is where the actual configuration values are provided. Each configuration value is required to refer to the accompanying definition such that tools

- can interpret the provided value correctly with respect to the definition of the value and
- can check whether the nature of the provided value matches the expectation set by the corresponding element on the definition side.

The concrete references from “value side” to “definition side” with the semantics of referring to the defining configuration element are

- `ApmcFunctionalClusterValue.definition`
- `ApmcContainerValue.definition`
- `ApmcContainerElementValue.definition`

This modeling aspect is depicted in Figure 3.4.

#### [constr\_10581] Existence of reference in the role `ApmcFunctionalClusterValue.definition`

Status: DRAFT

Imposition time: IT\_CfgFc

[For each `ApmcFunctionalClusterValue`, the reference in the role `definition` shall exist.]

**[constr\_10582] Existence of reference in the role `ApmcContainerValue.definition`**

Status: DRAFT

Imposition time: IT\_CfgFc

[For each `ApmcContainerValue`, the reference in the role `definition` shall exist.]

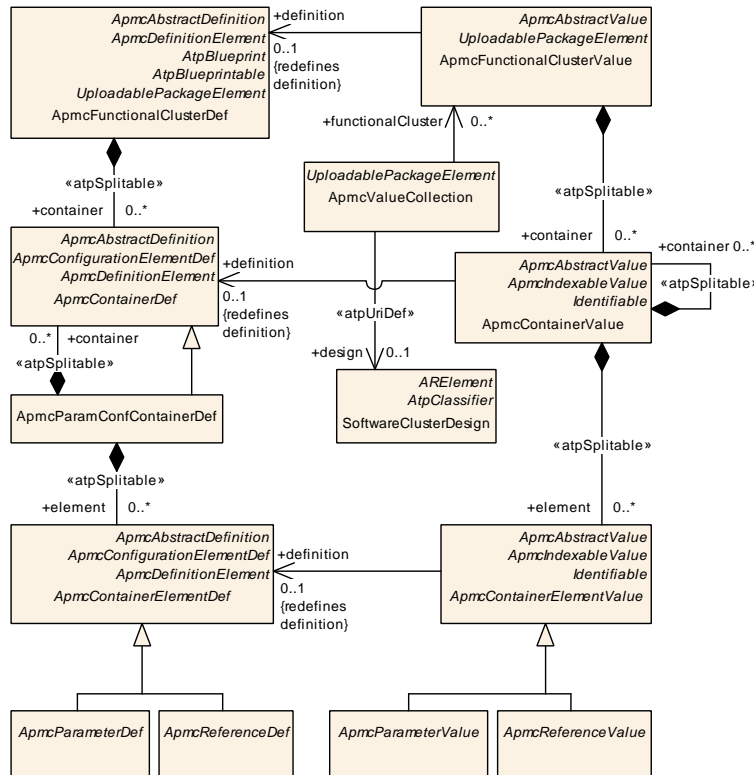


Figure 3.4: Overview of the `target-configuration` values

**[constr\_10583] Existence of reference in the role `ApmcContainerElementValue.definition`**

Status: DRAFT

Imposition time: IT\_CfgFc

[For each `ApmcContainerElementValue`, the reference in the role `definition` shall exist.]

**[AP\_TPS\_APMC\_01010] Root element on the “value side”**

Status: DRAFT

[The root element of the `target-configuration` on the “value side” is represented by meta-class `ApmcFunctionalClusterValue`.]

### [AP\_TPS\_APMC\_01036] Semantics of [ApmcValueCollection](#)

Status: DRAFT

[Meta-class [ApmcValueCollection](#) acts (as the name already suggests) like a collector, such that model elements that need to refer to a number of different [ApmcFunctionalClusterValues](#) can instead simply create one reference to the [ApmcValueCollection](#) that in turn refers to the affected [ApmcFunctionalClusterValues](#) in the role `functionalCluster`.]

<b>Class</b>	<b>ApmcValueCollection</b>			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::ValueSide			
<b>Note</b>	This class represents the entry point of the target-configuration on the "value" side. <b>Tags:</b> atp.Status=candidate atp.recommendedPackage=SoftwareClusterValues			
<b>Base</b>	<i>ARElement, ARObject, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable, UploadablePackageElement</i>			
<b>Aggregated by</b>	ARPackage.element			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
design	SoftwareClusterDesign	0..1	ref	This reference identifies the corresponding Software ClusterDesign. <b>Stereotypes:</b> atpUriDef <b>Tags:</b> atp.Status=candidate
functional Cluster	<a href="#">ApmcFunctionalClusterValue</a>	*	ref	This reference identifies the collection of functional clusters that are configured (partially) in the enclosing ApmcValueCollection. <b>Tags:</b> atp.Status=candidate

**Table 3.14: ApmcValueCollection**

<b>Class</b>	<b>ApmcFunctionalClusterValue</b>			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::ValueSide			
<b>Note</b>	This class represents the entry point into the configuration of a single functional cluster on the AUTOSAR adaptive platform on the "value" side. <b>Tags:</b> atp.Status=candidate atp.recommendedPackage=FunctionalClusterValues			
<b>Base</b>	<i>ARElement, ARObject, ApmcAbstractValue, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable, UploadablePackageElement</i>			
<b>Aggregated by</b>	ARPackage.element			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
container	<a href="#">ApmcContainerValue</a>	*	aggr	This aggregation represents the collection of containers aggregated directly by the functional cluster configuration definition. <b>Stereotypes:</b> atpSplitable <b>Tags:</b> atp.Splitkey=container.shortName atp.Status=candidate





Class		ApmcFunctionalClusterValue		
definition	<a href="#">ApmcFunctionalClusterDef</a>	0..1	ref	This reference implements the concrete relation of functional cluster configuration on the "value" side to functional cluster configuration on the "definition" side in the abstract structure of the machine configuration model.

**Table 3.15: ApmcFunctionalClusterValue**

### 3.3.1 Grouping of Configuration Values

The definition of grouping and nested structures of configuration elements, as described on the “definition side” in section 3.2.2 is reflected in the “value side” as well.

For this purpose, meta-class [ApmcContainerValue](#) has the ability to aggregate itself, see Figure 3.4.

Class		ApmcContainerValue		
Package	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::ValueSide			
Note	This meta-class represents a container on the "value" side of the target-configuration. <b>Tags:</b> atp.Status=candidate			
Base	<a href="#">ARObject</a> , <a href="#">ApmcAbstractValue</a> , <a href="#">ApmcIndexableValue</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
Aggregated by	<a href="#">ApmcContainerValue.container</a> , <a href="#">ApmcFunctionalClusterValue.container</a>			
Attribute	Type	Mult.	Kind	Note
container	<a href="#">ApmcContainerValue</a>	*	aggr	This aggregation represents the collection of sub-containers of the enclosing container. By using this aggregation arbitrary nested structures of containers can be created. <b>Stereotypes:</b> atpSplittable <b>Tags:</b> atp.Splitkey=container.shortName atp.Status=candidate
definition	<a href="#">ApmcContainerDef</a>	0..1	ref	This reference implements the concrete relation of container configuration on the "value" side to container configuration on the "definition" side in the abstract structure of the machine configuration model.
element	<a href="#">ApmcContainerElementValue</a>	*	aggr	This aggregation represents the collection of elements in the container. <b>Stereotypes:</b> atpSplittable <b>Tags:</b> atp.Splitkey=element.shortName atp.Status=candidate

**Table 3.16: ApmcContainerValue**

<b>Class</b>	<b>ApmcContainerElementValue</b> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::ValueSide			
<b>Note</b>	This abstract meta-class represents an element of a container on the "value" side of the target-configuration. <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	<i>ARObject</i> , <i>ApmcAbstractValue</i> , <i>ApmcIndexableValue</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
<b>Subclasses</b>	<i>ApmcParameterValue</i> , <i>ApmcReferenceValue</i>			
<b>Aggregated by</b>	<i>ApmcContainerValue</i> .element			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
definition	<a href="#">ApmcContainerElementDef</a>	0..1	ref	This reference implements the concrete relation of container element configuration on the "value" side to container element configuration on the "definition" side in the abstract structure of the machine configuration model. <b>Tags:</b> atp.Status=candidate

**Table 3.17: ApmcContainerElementValue**

### 3.3.2 Parameter

As explained in section 3.2.4, configuration parameter can be defined in different flavors. These flavors are also represented on the value side, as subclasses of the abstract base class [ApmcParameterValue](#).

Please find more information about the details in section 4.2.1.

<b>Class</b>	<b>ApmcParameterValue</b> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::ValueSide			
<b>Note</b>	This meta-class represents a parameter on the "value" side of the adaptive platform. The concrete nature of the parameter is clarified by a subclass. <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	<i>ARObject</i> , <i>ApmcAbstractValue</i> , <i>ApmcContainerElementValue</i> , <i>ApmcIndexableValue</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
<b>Subclasses</b>	<i>ApmcNumericalParamValue</i> , <i>ApmcTextualParamValue</i>			
<b>Aggregated by</b>	<i>ApmcContainerValue</i> .element			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 3.18: ApmcParameterValue**

### 3.3.3 Reference

References are also represented on the “value side” by means of abstract base class [ApmcReferenceValue](#). Please find more information about the details in section 4.2.2.

<b>Class</b>	<b><i>ApmcReferenceValue</i></b> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::ValueSide			
<b>Note</b>	This meta-class represents a reference on the "value" side of the adaptive platform. The concrete nature of the reference is clarified by a subclass. <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	<i>ARObject</i> , <a href="#">ApmcAbstractValue</a> , <a href="#">ApmcContainerElementValue</a> , <a href="#">ApmcIndexableValue</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Subclasses</b>	<a href="#">ApmcAbstractInstanceReferenceValue</a> , <a href="#">ApmcAbstractReferenceValue</a> , <a href="#">ApmcContainerReferenceValue</a>			
<b>Aggregated by</b>	<a href="#">ApmcContainerValue.element</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 3.19: ApmcReferenceValue**



## 4 Details

This chapter provides a description of the configuration elements with a focus on the details.

### 4.1 Definition Side

This section contains a detailed description of configuration elements on the “definition side”.

#### 4.1.1 Parameter

The detailed definition of the modeling of parameter definitions is depicted in Figure 4.1.

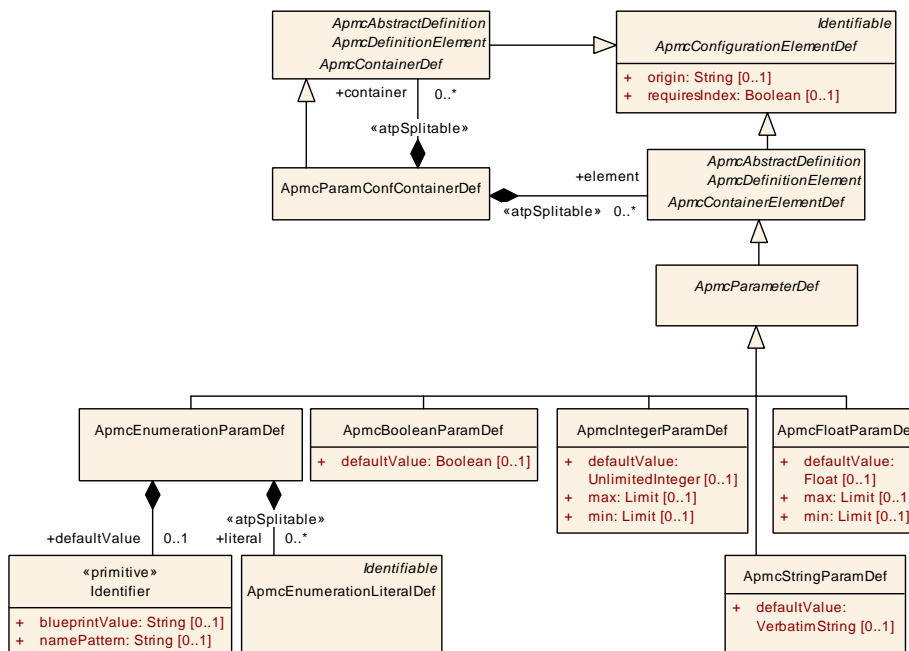


Figure 4.1: Overview of the manifest parameter definition

##### 4.1.1.1 Enumeration Parameter

#### [AP\_TPS\_APMC\_01011] Definition of an enumeration-style parameter

Status: DRAFT

[The definition of an enumeration-style parameter is done by means of meta-classes [ApmcEnumerationParamDef](#) and [ApmcEnumerationLiteralDef](#). The former

class is used to model the enumeration itself, while the latter class is used to model the specific values that the enumeration can take.

The literals that the enumeration can take are aggregated in the role `literal`.]

### [AP\_TPS\_APMC\_01012] Definition of default value for enumeration-style parameter

*Status:* DRAFT

[It is possible to define a default value for an `ApmcEnumerationParamDef`, such that any parameter value that is created with a reference in the role `definition` can be created and immediately gets assigned a value taken from `ApmcEnumerationParamDef.defaultValue`.

The goal of the definition of a default value is to support a creation of a number of parameter values at a time that each are automatically assigned a value that reasonably approximates the final setting of the parameter in the sense that the user is “not wrong” if the parameter value is kept at the default value for the time being.

Of course, in the majority of cases, the default value will not remain and it will, at some point, be replaced by another value that is the result of the user’s informed decision.]

Please note that [`constr_10585`] also applies for the definition of the `ApmcEnumerationParamDef.defaultValue`.

Note further that the class that corresponds to `ApmcEnumerationParamDef` on the “value side” is `ApmcTextualParamValue`.

<b>Class</b>	<b>ApmcEnumerationParamDef</b>			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::DefinitionSide			
<b>Note</b>	This meta-class represents an enumeration parameter on the "definition" side of the target-configuration. <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	<i>ARObject, ApmcAbstractDefinition, ApmcConfigurationElementDef, ApmcContainerElementDef, ApmcDefinitionElement, ApmcParameterDef, Identifiable, MultilanguageReferrable, Referrable</i>			
<b>Aggregated by</b>	<code>ApmcParamConfContainerDef.element</code>			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
defaultValue	Identifier	0..1	attr	This attribute represents the definition of the default value for the enclosing parameter. The default value applies if the parameter is instantiated on the "value" side without assigning a dedicated value to it. <b>Tags:</b> atp.Status=candidate
literal	<code>ApmcEnumerationLiteralDef</code>	*	aggr	This aggregation represents the collection of enumerators that are part of the enumeration. <b>Stereotypes:</b> atpSplittable <b>Tags:</b> atp.Splitkey=literal.shortName atp.Status=candidate

**Table 4.1: ApmcEnumerationParamDef**

<b>Class</b>	<b>ApmcEnumerationLiteralDef</b>			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::DefinitionSide			
<b>Note</b>	This meta-class represents an enumerator (or enumeration literal) on the "definition" side of the target-configuration. <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	<i>ARObject</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
<b>Aggregated by</b>	<a href="#">ApmcEnumerationParamDef.literal</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 4.2: ApmcEnumerationLiteralDef**

#### 4.1.1.2 Boolean Parameter

Meta-class [ApmcBooleanParamDef](#) is used to define a configuration parameter that carries a boolean value.

The class that corresponds to [ApmcBooleanParamDef](#) on the “value side” is [ApmcNumericalParamValue](#).

For [ApmcBooleanParamDef](#), it is possible to define a `defaultValue`, similar to [AP\_TPS\_APMC\_01012]. The values used for the definition of the `defaultValue` shall follow the regulation imposed by [constr\_10584].

<b>Class</b>	<b>ApmcBooleanParamDef</b>			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::DefinitionSide			
<b>Note</b>	This meta-class represents a boolean parameter on the "definition" side of the target-configuration. <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	<i>ARObject</i> , <i>ApmcAbstractDefinition</i> , <i>ApmcConfigurationElementDef</i> , <i>ApmcContainerElementDef</i> , <i>ApmcDefinitionElement</i> , <i>ApmcParameterDef</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
<b>Aggregated by</b>	<a href="#">ApmcParamConfContainerDef.element</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
defaultValue	Boolean	0..1	attr	This attribute represents the definition of the default value for the enclosing parameter. The default value applies if the parameter is instantiated on the "value" side without assigning a dedicated value to it. <b>Tags:</b> atp.Status=candidate

**Table 4.3: ApmcBooleanParamDef**

#### 4.1.1.3 Integer Parameter

Meta-class [ApmcIntegerParamDef](#) is used to define a configuration parameter that carries an integer value.

The class that corresponds to [ApmcIntegerParamDef](#) on the “value side” is [ApmcNumericalParamValue](#).

For [ApmcIntegerParamDef](#), it is possible to define a `defaultValue`.

It is also possible to define the allowed range of the specific [ApmcIntegerParamDef](#) by means of attributes `min` and `max`.

<b>Class</b>	<b>ApmcIntegerParamDef</b>			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::DefinitionSide			
<b>Note</b>	This meta-class represents an integer parameter on the "definition" side of the target-configuration. <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	<a href="#">ARObject</a> , <a href="#">ApmcAbstractDefinition</a> , <a href="#">ApmcConfigurationElementDef</a> , <a href="#">ApmcContainerElementDef</a> , <a href="#">ApmcDefinitionElement</a> , <a href="#">ApmcParameterDef</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Aggregated by</b>	<a href="#">ApmcParamConfContainerDef</a> .element			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
defaultValue	UnlimitedInteger	0..1	attr	This attribute represents the definition of the default value for the enclosing parameter. The default value applies if the parameter is instantiated on the "value" side without assigning a dedicated value to it. <b>Tags:</b> atp.Status=candidate
max	Limit	0..1	attr	This attribute defines the upper limit of the parameter. This means that the value of the parameter shall never be greater than the defined upper limit. <b>Tags:</b> atp.Status=candidate
min	Limit	0..1	attr	This attribute defines the lower limit of the parameter. This means that the value of the parameter shall never be less than the defined lower limit. <b>Tags:</b> atp.Status=candidate

**Table 4.4: ApmcIntegerParamDef**

#### 4.1.1.4 Float Parameter

Meta-class [ApmcFloatParamDef](#) is used to define a configuration parameter that carries a floating-point value.

The class that corresponds to [ApmcFloatParamDef](#) on the "value side" is [ApmcNumericalParamValue](#).

For [ApmcFloatParamDef](#), it is possible to define a `defaultValue`.

It is also possible to define the allowed range of the specific [ApmcFloatParamDef](#) by means of attributes `min` and `max`.

<b>Class</b>	<b>ApmcFloatParamDef</b>			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::DefinitionSide			
<b>Note</b>	This meta-class represents a float parameter on the "definition" side of the target-configuration. <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	<a href="#">ARObject</a> , <a href="#">ApmcAbstractDefinition</a> , <a href="#">ApmcConfigurationElementDef</a> , <a href="#">ApmcContainerElementDef</a> , <a href="#">ApmcDefinitionElement</a> , <a href="#">ApmcParameterDef</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Aggregated by</b>	<a href="#">ApmcParamConfContainerDef</a> .element			





Class		ApmcFloatParamDef		
Attribute	Type	Mult.	Kind	Note
defaultValue	Float	0..1	attr	This attribute represents the definition of the default value for the enclosing parameter. The default value applies if the parameter is instantiated on the "value" side without assigning a dedicated value to it. <b>Tags:</b> atp.Status=candidate
max	Limit	0..1	attr	This attribute defines the upper limit of the parameter. This means that the value of the parameter shall never be greater than the defined upper limit. <b>Tags:</b> atp.Status=candidate
min	Limit	0..1	attr	This attribute defines the lower limit of the parameter. This means that the value of the parameter shall never be less than the defined lower limit. <b>Tags:</b> atp.Status=candidate

**Table 4.5: ApmcFloatParamDef**

#### 4.1.1.5 String Parameter

Class		ApmcStringParamDef		
Package	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::DefinitionSide			
Note	This meta-class represents an string parameter on the "definition" side of the target-configuration. <b>Tags:</b> atp.Status=candidate			
Base	<a href="#">ARObject</a> , <a href="#">ApmcAbstractDefinition</a> , <a href="#">ApmcConfigurationElementDef</a> , <a href="#">ApmcContainerElementDef</a> , <a href="#">ApmcDefinitionElement</a> , <a href="#">ApmcParameterDef</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
Aggregated by	<a href="#">ApmcParamConfContainerDef</a> .element			
Attribute	Type	Mult.	Kind	Note
defaultValue	VerbatimString	0..1	attr	This attribute represents the definition of the default value for the enclosing parameter. The default value applies if the parameter is instantiated on the "value" side without assigning a dedicated value to it. <b>Tags:</b> atp.Status=candidate

**Table 4.6: ApmcStringParamDef**

Meta-class [ApmcStringParamDef](#) is used to define a configuration parameter that carries a string value.

The class that corresponds to [ApmcStringParamDef](#) on the "value side" is [ApmcTextualParamValue](#).

For [ApmcStringParamDef](#), it is possible to define a [defaultValue](#).

#### 4.1.2 Reference

The detailed modeling of references on the definition side is depicted in Figure 4.2. The modeling supports three kinds of references:

- Reference to container (see section 4.1.2.1)
- Reference to element of the M2 model (so-called *foreign* reference, see section 4.1.2.2)
- Instance Reference to element of the M2 model (so-called *instanceRef*, see section 4.1.2.3)

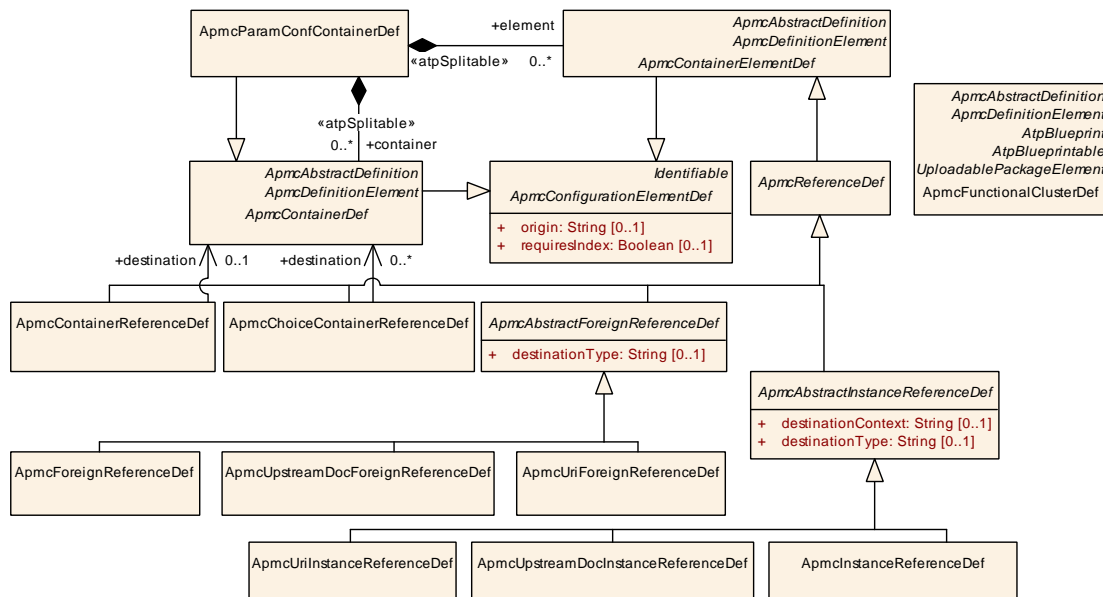


Figure 4.2: Overview of the manifest reference definition

#### 4.1.2.1 Reference to Container

Within the M1 model, it is not possible to reference elements represented by abstract meta-class `ApmcContainerElementDef`. The only valid target of a reference is an instance of abstract base class `ApmcContainerDef`.

#### [AP\_TPS\_APMC\_01014] Semantics of meta-class `ApmcContainerReferenceDef`

Status: DRAFT

[Meta-class `ApmcContainerReferenceDef` shall be used if a reference to one `ApmcContainerDef` needs to be modeled.

The usage of meta-class `ApmcContainerReferenceDef` is independent of whether the target of the reference is defined in the scope of the same `ApmcFunctionalClusterDef` or not.]

### [AP\_TPS\_APMC\_01015] Semantics of reference [ApmcContainerReferenceDef.destination](#)

Status: DRAFT

[Attribute [ApmcContainerReferenceDef.destination](#) is supposed to be filled with a fully-qualified [shortName](#) of the reference target.]

<b>Class</b>	<b>ApmcContainerReferenceDef</b>			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::DefinitionSide			
<b>Note</b>	This meta-class represents a reference to another container of the target-configuration. <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	<a href="#">ARObject</a> , <a href="#">ApmcAbstractDefinition</a> , <a href="#">ApmcConfigurationElementDef</a> , <a href="#">ApmcContainerElementDef</a> , <a href="#">ApmcDefinitionElement</a> , <a href="#">ApmcReferenceDef</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Aggregated by</b>	<a href="#">ApmcParamConfContainerDef.element</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
destination	<a href="#">ApmcContainerDef</a>	0..1	ref	This reference identifies the target container. <b>Tags:</b> atp.Status=candidate

**Table 4.7: ApmcContainerReferenceDef**

### [AP\_TPS\_APMC\_01016] Semantics of meta-class [ApmcChoiceContainerReferenceDef](#)

Status: DRAFT

[Meta-class [ApmcContainerReferenceDef](#) shall be used if a reference to more than one [ApmcContainerDef](#) needs to be modeled.

The semantics of the collection of references is that finally only a single reference out of the possible references shall exist on the “value side”.

The usage of meta-class [ApmcChoiceContainerReferenceDef](#) is independent of whether the target of the reference is defined in the scope of the same [ApmcFunctionalClusterDef](#) or not.]

<b>Class</b>	<b>ApmcChoiceContainerReferenceDef</b>			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::DefinitionSide			
<b>Note</b>	This meta-class represents the ability to establish a reference that points to a collection of targets, out of which only one element can be picked at any given time. <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	<a href="#">ARObject</a> , <a href="#">ApmcAbstractDefinition</a> , <a href="#">ApmcConfigurationElementDef</a> , <a href="#">ApmcContainerElementDef</a> , <a href="#">ApmcDefinitionElement</a> , <a href="#">ApmcReferenceDef</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Aggregated by</b>	<a href="#">ApmcParamConfContainerDef.element</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>





Class	ApmcChoiceContainerReferenceDef			
destination	<a href="#">ApmcContainerDef</a>	*	ref	This represents the collection of potential reference targets. <b>Tags:</b> atp.Status=candidate

**Table 4.8: ApmcChoiceContainerReferenceDef**

#### 4.1.2.2 Foreign Reference

The modeling of a “foreign”<sup>1</sup> reference is based on the definition of the abstract base-class [ApmcAbstractForeignReferenceDef](#).

#### [AP\_TPS\_APMC\_01017] Semantics of attribute [ApmcAbstractForeignReferenceDef.destinationType](#)

*Status:* DRAFT

[Attribute [ApmcAbstractForeignReferenceDef.destinationType](#) is supposed to be filled with an Identification of the class of the reference target.

The class name is suppose to be provided in “schema notation” (Example: use PROCESS-DESIGN as a representation of meta-class [ProcessDesign](#)).]

Class	ApmcAbstractForeignReferenceDef (abstract)			
Package	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::DefinitionSide			
Note	This abstract meta-class represents the ability to establish a flat (as opposed to an instanceRef) reference to model elements out of the context of target-configuration, i.e. in the design model. <b>Tags:</b> atp.Status=candidate			
Base	<a href="#">ARObject</a> , <a href="#">ApmcAbstractDefinition</a> , <a href="#">ApmcConfigurationElementDef</a> , <a href="#">ApmcContainerElementDef</a> , <a href="#">ApmcDefinitionElement</a> , <a href="#">ApmcReferenceDef</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
Subclasses	<a href="#">ApmcForeignReferenceDef</a> , <a href="#">ApmcUpstreamDocForeignReferenceDef</a> , <a href="#">ApmcUriForeignReferenceDef</a>			
Aggregated by	<a href="#">ApmcParamConfContainerDef.element</a>			
Attribute	Type	Mult.	Kind	Note
destinationType	String	0..1	attr	This attribute carries the identification of the target's class. This way, the possible reference targets can be restricted on the "definition" side of the target-configuration. This is necessary to preserve the intended semantics of the foreign reference. <b>Tags:</b> atp.Status=candidate

**Table 4.9: ApmcAbstractForeignReferenceDef**

<sup>1</sup>in the sense that the target of the reference is not a part of the M1 model



### [AP\_TPS\_APMC\_01018] Semantics of [ApmcForeignReferenceDef](#)

Status: DRAFT

[Meta-class [ApmcForeignReferenceDef](#) is used to establish a **flat** reference to a referable<sup>2</sup> model-element on the M2 level.

In the case of the [target-configuration](#), the M2 model elements are used to define design-related model content and therefore the [ApmcForeignReferenceDef](#) is factually used to **establish a reference from the configuration domain to the design domain**.

The target of the reference is expected to exist in the same model scope as the owner of the [ApmcForeignReferenceDef](#).]

<b>Class</b>	<a href="#">ApmcForeignReferenceDef</a>			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::DefinitionSide			
<b>Note</b>	This meta-class represents the ability to establish a flat (as opposed to an instanceRef) reference to model elements out of the context of target-configuration, i.e. reference the element in the design model that contributes to the semantics of the target-configuration model. <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	<a href="#">ARObject</a> , <a href="#">ApmcAbstractDefinition</a> , <a href="#">ApmcAbstractForeignReferenceDef</a> , <a href="#">ApmcConfigurationElementDef</a> , <a href="#">ApmcContainerElementDef</a> , <a href="#">ApmcDefinitionElement</a> , <a href="#">ApmcReferenceDef</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Aggregated by</b>	<a href="#">ApmcParamConfContainerDef</a> .element			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

Table 4.10: [ApmcForeignReferenceDef](#)

### [AP\_TPS\_APMC\_01019] Semantics of meta-class [ApmcUriForeignReferenceDef](#)

Status: DRAFT

[In principle, a [ApmcUriForeignReferenceDef](#) fulfills the same role as a [ApmcForeignReferenceDef](#).

The difference is that in the case of the [ApmcUriForeignReferenceDef](#), the target of the **flat** reference is **not** required to exist in the same model scope.

On the “value side”, if the target of the reference does not exist in the same model scope, then the existence of the [ApmcUriForeignReferenceDef](#) can serve as an **additional identifying element** of the enclosing [ApmcContainerValue](#) of the [ApmcUriForeignReferenceDef](#).]

<sup>2</sup>i.e. derived from meta-class [Referrable](#)

<b>Class</b>	<b>ApmcUriForeignReferenceDef</b>			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::DefinitionSide			
<b>Note</b>	This meta-class represents the ability to establish a flat (as opposed to an instanceRef) reference to model elements out of the context of target-configuration.  The target of the reference is not required to exist. The reference itself is already considered an identifying element.  <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	<a href="#">ARObject</a> , <a href="#">ApmcAbstractDefinition</a> , <a href="#">ApmcAbstractForeignReferenceDef</a> , <a href="#">ApmcConfigurationElementDef</a> , <a href="#">ApmcContainerElementDef</a> , <a href="#">ApmcDefinitionElement</a> , <a href="#">ApmcReferenceDef</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Aggregated by</b>	<a href="#">ApmcParamConfContainerDef</a> .element			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 4.11: ApmcUriForeignReferenceDef**

### [AP\_TPS\_APMC\_01020] Semantics of meta-class [ApmcUpstreamDocForeignReferenceDef](#)

*Status:* DRAFT

[The purpose of the [ApmcUpstreamDocForeignReferenceDef](#) is to support the creation of a **flat** reference to a model **element in the design domain from which the enclosing container was created by derivation**.

The existence of [ApmcUpstreamDocForeignReferenceValue](#) that refers to a [ApmcUpstreamDocForeignReferenceDef](#) in the role *definition* is considered as pure meta-data.

In contrast to [ApmcForeignReferenceDef](#) and [ApmcUriForeignReferenceDef](#), the [ApmcUpstreamDocForeignReferenceValue](#) does not carry relevant model semantics and serves only as a protocol of how the enclosing container came into existence.]

<b>Class</b>	<b>ApmcUpstreamDocForeignReferenceDef</b>			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::DefinitionSide			
<b>Note</b>	This meta-class represents the ability to establish a flat (as opposed to an instanceRef) reference to model elements out of the context of target-configuration, i.e. reference the element in the design model that sparked the existence of the owner of the reference.  <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	<a href="#">ARObject</a> , <a href="#">ApmcAbstractDefinition</a> , <a href="#">ApmcAbstractForeignReferenceDef</a> , <a href="#">ApmcConfigurationElementDef</a> , <a href="#">ApmcContainerElementDef</a> , <a href="#">ApmcDefinitionElement</a> , <a href="#">ApmcReferenceDef</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Aggregated by</b>	<a href="#">ApmcParamConfContainerDef</a> .element			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 4.12: ApmcUpstreamDocForeignReferenceDef**

### 4.1.2.3 Instance Reference

The instance references described in this chapter support the structured referencing to a model tree in which the AUTOSAR type-prototype is implemented. These instance references will be realized by meta-class `AnyInstanceRef` in an AUTOSAR model.

#### [AP\_TPS\_APMC\_01021] Semantics of attribute `ApmcAbstractInstanceReferenceDef.destinationContext`

Status: DRAFT

[The attribute `destinationContext` of meta-class `ApmcAbstractInstanceReferenceDef` establishes the collection of context references used to define the `ApmcAbstractInstanceReferenceDef`.]

#### [AP\_TPS\_APMC\_01022] Semantics of attribute `ApmcAbstractInstanceReferenceDef.destinationType`

Status: DRAFT

[The attribute `destinationType` of meta-class `ApmcAbstractInstanceReferenceDef` establishes the target reference used to define the `ApmcAbstractInstanceReferenceDef`.]

<b>Class</b>	<code>ApmcAbstractInstanceReferenceDef</code> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::DefinitionSide			
<b>Note</b>	This abstract meta-class represents the ability to establish a <code>instanceRef</code> (as opposed to flat) reference to model elements. Subclasses clarify the specific semantics. <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	<code>ARObject</code> , <code>ApmcAbstractDefinition</code> , <code>ApmcConfigurationElementDef</code> , <code>ApmcContainerElementDef</code> , <code>ApmcDefinitionElement</code> , <code>ApmcReferenceDef</code> , <code>Identifiable</code> , <code>MultilanguageReferrable</code> , <code>Referrable</code>			
<b>Subclasses</b>	<code>ApmcInstanceReferenceDef</code> , <code>ApmcUpstreamDocInstanceReferenceDef</code> , <code>ApmcUriInstanceReferenceDef</code>			
<b>Aggregated by</b>	<code>ApmcParamConfContainerDef.element</code>			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
<code>destinationContext</code>	String	0..1	attr	This attribute carries the ordered list (separated by whitespace) of identification of context classes. This way, the possible reference targets can be restricted on the "definition" side of the target-configuration. This is necessary to preserve the intended semantics of the instance reference. <b>Tags:</b> atp.Status=candidate
<code>destinationType</code>	String	0..1	attr	This attribute carries the identification of the target's class. This way, the possible reference targets can be restricted on the "definition" side of the target-configuration. This is necessary to preserve the intended semantics of the instance reference. <b>Tags:</b> atp.Status=candidate

**Table 4.13: `ApmcAbstractInstanceReferenceDef`**

### [AP\_TPS\_APMC\_01023] Semantics of [ApmcInstanceReferenceDef](#)

Status: DRAFT

[Meta-class [ApmcInstanceReferenceDef](#) is used to establish a **structured** reference to a referable<sup>3</sup> model-element on the M2 level.

In the case of the [target-configuration](#), the M2 model elements are used to define design-related model content and therefore the [ApmcInstanceReferenceDef](#) is factually used to **establish a reference from the configuration domain to the design domain**.

All context references and the target reference are expected to exist in the same model scope as the owner of the [ApmcInstanceReferenceDef](#).]

<b>Class</b>	<b>ApmcInstanceReferenceDef</b>			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::DefinitionSide			
<b>Note</b>	This meta-class represents the ability to establish a instanceRef (as opposed to flat) reference to model elements out of the context of target-configuration, i.e. in the design model. <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	<a href="#">ARObject</a> , <a href="#">ApmcAbstractDefinition</a> , <a href="#">ApmcAbstractInstanceReferenceDef</a> , <a href="#">ApmcConfigurationElementDef</a> , <a href="#">ApmcContainerElementDef</a> , <a href="#">ApmcDefinitionElement</a> , <a href="#">ApmcReferenceDef</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Aggregated by</b>	<a href="#">ApmcParamConfContainerDef.element</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

Table 4.14: [ApmcInstanceReferenceDef](#)

### [AP\_TPS\_APMC\_01024] Semantics of meta-class [ApmcUriInstanceReferenceDef](#)

Status: DRAFT

[In principle, a [ApmcUriInstanceReferenceDef](#) fulfills the same role as a [ApmcInstanceReferenceDef](#).

The difference is that in the case of the [ApmcUriInstanceReferenceDef](#) the target of the **structured** reference is **not** required to exist in the same model scope.

On the “value side”, if the target of the reference does not exist in the same model scope, then the existence of the [ApmcUriInstanceReferenceDef](#) can serve as an **additional identifying element** of the enclosing [ApmcContainerValue](#) of the [ApmcUriInstanceReferenceDef](#).]

<sup>3</sup>i.e. derived from meta-class [Referrable](#)

<b>Class</b>	<b>ApmcUriInstanceReferenceDef</b>			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::DefinitionSide			
<b>Note</b>	This meta-class represents the ability to establish a flat (as opposed to an instanceRef) reference to model elements on design level. In other words, it is possible to annotate which design element was taken as the basis for the target-configuration model element that owns the reference. The target of the reference is not required to exist. The reference itself is already considered an identifying element. <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	<a href="#">ARObject</a> , <a href="#">ApmcAbstractDefinition</a> , <a href="#">ApmcAbstractInstanceReferenceDef</a> , <a href="#">ApmcConfigurationElementDef</a> , <a href="#">ApmcContainerElementDef</a> , <a href="#">ApmcDefinitionElement</a> , <a href="#">ApmcReferenceDef</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Aggregated by</b>	<a href="#">ApmcParamConfContainerDef</a> .element			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 4.15: ApmcUriInstanceReferenceDef**

<b>Class</b>	<b>ApmcUpstreamDocInstanceReferenceDef</b>			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::DefinitionSide			
<b>Note</b>	This meta-class represents the ability to establish a flat (as opposed to an instanceRef) reference to model elements on design level in order to identify a model element that sparked the existence of the owner of the reference. In other words, it is possible to annotate which design element was taken as the basis for the target-configuration model element that owns the reference. <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	<a href="#">ARObject</a> , <a href="#">ApmcAbstractDefinition</a> , <a href="#">ApmcAbstractInstanceReferenceDef</a> , <a href="#">ApmcConfigurationElementDef</a> , <a href="#">ApmcContainerElementDef</a> , <a href="#">ApmcDefinitionElement</a> , <a href="#">ApmcReferenceDef</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Aggregated by</b>	<a href="#">ApmcParamConfContainerDef</a> .element			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 4.16: ApmcUpstreamDocInstanceReferenceDef**

## [AP\_TPS\_APMC\_01025] Semantics of meta-class [ApmcUpstreamDocInstanceReferenceValue](#)

*Status:* DRAFT

[The purpose of the [ApmcUpstreamDocInstanceReferenceValue](#) is to establish a **structured** reference to a model **element in the design domain from which the enclosing container was created by derivation**.

The existence of [ApmcUpstreamDocInstanceReferenceValue](#) is considered as pure meta-data.

In contrast to [ApmcInstanceReferenceDef](#) and [ApmcUriInstanceReferenceDef](#), the [ApmcUpstreamDocInstanceReferenceValue](#) does not carry relevant model semantics and serves only as a protocol of how the enclosing container came into existence.]

## 4.2 Value Side

This section contains a detailed description of configuration elements on the “value side”.

### 4.2.1 Parameter

#### [AP\_TPS\_APMC\_01026] Semantics of attribute `ApmcIndexableValue.index`

Status: DRAFT

[The value of the attribute `ApmcIndexableValue.index` identifies the index of the element in a collection for which on the “definition side” the attribute `ApmcConfigurationElementDef.requiresIndex` has been set.]

#### [AP\_TPS\_APMC\_01027] Gaps in the value range of an ordered collection

Status: DRAFT

[The values of the individual `index` attributes within an ordered collection may have gaps. The ordering is not based on the usage of every single index value in the affected interval, but on the numerical sorting of the index values.]

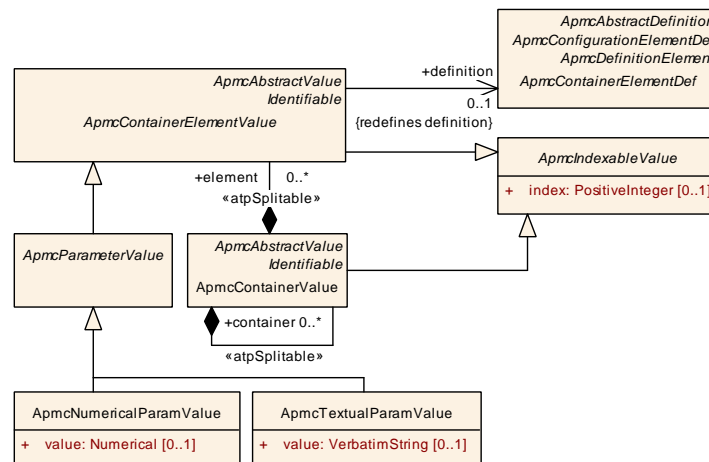


Figure 4.3: Overview of the manifest parameter values

<b>Class</b>	<i>ApmcIndexableValue</i> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::ValueSide			
<b>Note</b>	This abstract meta-class provides the ability to define model elements that carry an index on the "value" side of the target-configuration.			
<b>Base</b>	<i>ARObject</i>			
<b>Subclasses</b>	<i>ApmcContainerElementValue</i> , <i>ApmcContainerValue</i>			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>





Class	ApmcIndexableValue (abstract)			
index	PositiveInteger	0..1	attr	This attribute represents the index value of an element of a collection where the "logical" position of the element in the collection carries a dedicated semantics

**Table 4.17: ApmcIndexableValue**

#### 4.2.1.1 Numerical Parameter Value

**[constr\_10584] Allowed value for a [ApmcNumericalParamValue](#) that implements an boolean-style parameter**

Status: DRAFT

Imposition time: IT\_CfgFc

[For any [ApmcNumericalParamValue](#) that refers to an [ApmcBooleanParamDef](#) in the role *definition*, the content of the [ApmcNumericalParamValue.value](#) shall only be either 0 or 1.]

Class	ApmcNumericalParamValue			
Package	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::ValueSide			
Note	This meta-class represents a numerical parameter on the "value" side of the target-configuration. Tags: atp.Status=candidate			
Base	<a href="#">ARObject</a> , <a href="#">ApmcAbstractValue</a> , <a href="#">ApmcContainerElementValue</a> , <a href="#">ApmcIndexableValue</a> , <a href="#">ApmcParameter Value</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
Aggregated by	<a href="#">ApmcContainerValue.element</a>			
Attribute	Type	Mult.	Kind	Note
value	Numerical	0..1	attr	This attribute represents the value of the numerical parameter. Tags: atp.Status=candidate

**Table 4.18: ApmcNumericalParamValue**

#### 4.2.1.2 Textual Parameter Value

According to its nature of being an enumeration-style parameter, any [ApmcEnumerationParamDef](#) shall not accept any value that is not represented by any of the aggregated [ApmcEnumerationLiteralDefs](#). If it is intended to allow for assigning an arbitrary string value to the parameter, an [ApmcStringParamDef](#) should be preferred.





### [AP\_TPS\_APMC\_01037] Semantics of meta-class [ApmcContainerReferenceValue](#)

Status: DRAFT

[Meta-class [ApmcContainerReferenceValue](#) provides the ability to refer to another [ApmcContainerValue](#).

The target of this reference is required to exist in the scope of the M1 model, but does not necessarily have to be owned by the same [ApmcFunctionalClusterValue](#) that also owns the [ApmcContainerValue](#) in which said [ApmcContainerReferenceValue](#) is aggregated.]

<b>Class</b>	<b>ApmcContainerReferenceValue</b>			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::ValueSide			
<b>Note</b>	This meta-class represents a reference to another container of the target-configuration. <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	<a href="#">ARObject</a> , <a href="#">ApmcAbstractValue</a> , <a href="#">ApmcContainerElementValue</a> , <a href="#">ApmcIndexableValue</a> , <a href="#">ApmcReferenceValue</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Aggregated by</b>	<a href="#">ApmcContainerValue</a> .element			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
value	<a href="#">ApmcContainerValue</a>	0..1	ref	This reference identifies the target container. <b>Tags:</b> atp.Status=candidate

Table 4.20: [ApmcContainerReferenceValue](#)

### [AP\_TPS\_APMC\_01028] Semantics of [ApmcAbstractReferenceValue](#)

Status: DRAFT

[Abstract meta-class [ApmcAbstractReferenceValue](#) is used as a base class for the creation of a **flat** reference to an element of the M2 design model.

In other words, all subclasses of [ApmcAbstractReferenceValue](#) are used to establish a reference from the configuration model to the design model on the “value side”.]

The modeling of [ApmcAbstractReferenceValue](#) and its sub-classes is sketched in Figure 4.5.

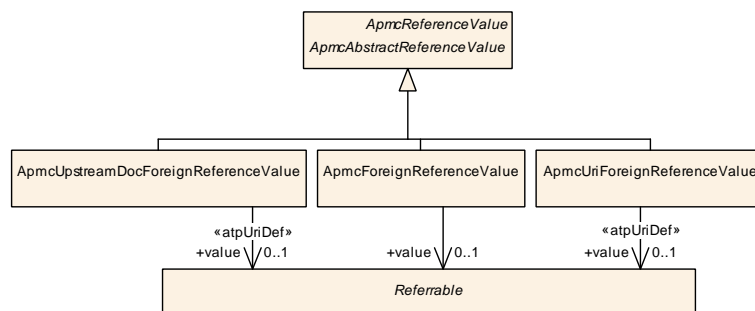


Figure 4.5: Overview of the manifest foreign reference values

<b>Class</b>	<b>ApmcAbstractReferenceValue</b> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::ValueSide			
<b>Note</b>	This abstract meta-class represents the ability to establish a flat (as opposed to an instanceRef) reference to model elements out of the context of target-configuration. <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	ARObject, <a href="#">ApmcAbstractValue</a> , <a href="#">ApmcContainerElementValue</a> , <a href="#">ApmcIndexableValue</a> , <a href="#">ApmcReferenceValue</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Subclasses</b>	<a href="#">ApmcForeignReferenceValue</a> , <a href="#">ApmcUpstreamDocForeignReferenceValue</a> , <a href="#">ApmcUriForeignReferenceValue</a>			
<b>Aggregated by</b>	<a href="#">ApmcContainerValue.element</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 4.21: ApmcAbstractReferenceValue**

### [AP\_TPS\_APMC\_01029] Semantics of meta-class [ApmcForeignReferenceValue](#)

*Status:* DRAFT

[Meta-class [ApmcForeignReferenceValue](#) provides the ability to refer to a referable element of the M2 model by means of the attribute [value](#).

The target of this reference is required to exist in the scope of the M2 model.]

<b>Class</b>	<b>ApmcForeignReferenceValue</b>			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::ValueSide			
<b>Note</b>	This meta-class represents the ability to establish a flat (as opposed to an instanceRef) reference to model elements out of the context of target-configuration, i.e. in the design model. <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	ARObject, <a href="#">ApmcAbstractReferenceValue</a> , <a href="#">ApmcAbstractValue</a> , <a href="#">ApmcContainerElementValue</a> , <a href="#">ApmcIndexableValue</a> , <a href="#">ApmcReferenceValue</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Aggregated by</b>	<a href="#">ApmcContainerValue.element</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
value	<a href="#">Referrable</a>	0..1	ref	This reference identifies the target model element that exists outside the target-configuration, i.e. in the design model. <b>Tags:</b> atp.Status=candidate

**Table 4.22: ApmcForeignReferenceValue**

### [AP\_TPS\_APMC\_01030] Semantics of meta-class [ApmcUriForeignReferenceValue](#)

*Status:* DRAFT

[Meta-class [ApmcUriForeignReferenceValue](#) provides the ability to refer to a referable element of the M2 model by means of the attribute [value](#).

The target of this reference is **not** required to exist in the scope of the M2 model. If the target does not exist, then the existence of the reference is taken as an identifying

characteristic of the [ApmcContainerValue](#) that owns the [ApmcUriForeignReferenceValue](#).]

<b>Class</b>	<b>ApmcUriForeignReferenceValue</b>			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::ValueSide			
<b>Note</b>	This meta-class represents the ability to establish a flat (as opposed to an instanceRef) reference to model elements out of the context of target-configuration, i.e. in the design model.  The referenced model element does not need to exist in the current model scope. The reference itself carries a semantics of being an identifying factor, even without the target element to exist.  <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	<a href="#">ARObject</a> , <a href="#">ApmcAbstractReferenceValue</a> , <a href="#">ApmcAbstractValue</a> , <a href="#">ApmcContainerElementValue</a> , <a href="#">ApmcIndexableValue</a> , <a href="#">ApmcReferenceValue</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Aggregated by</b>	<a href="#">ApmcContainerValue</a> .element			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
value	<a href="#">Referrable</a>	0..1	ref	This reference identifies the target model element that exists outside the target-configuration, i.e. in the design model.  The referenced model element does not need to exist in the current model scope. The reference itself carries a semantics of being an identifying factor, even without the target element to exist.  <b>Stereotypes:</b> atpUriDef <b>Tags:</b> atp.Status=candidate

**Table 4.23: ApmcUriForeignReferenceValue**

### [AP\_TPS\_APMC\_01031] Semantics of meta-class [ApmcUpstreamDocForeignReferenceValue](#)

*Status:* DRAFT

[The purpose of the [ApmcUpstreamDocForeignReferenceValue](#) is to establish a **flat** reference to a **model element in the design domain from which the enclosing container was created by derivation**.

The existence of [ApmcUpstreamDocForeignReferenceValue](#) is considered as pure meta-data.

In contrast to [ApmcForeignReferenceDef](#) and [ApmcUriForeignReferenceDef](#), the [ApmcUpstreamDocForeignReferenceValue](#) does not carry relevant model semantics and serves only as a protocol of how the enclosing container came into existence.]

<b>Class</b>	<b>ApmcUpstreamDocForeignReferenceValue</b>			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::ValueSide			
<b>Note</b>	This meta-class represents the ability to establish a flat (as opposed to an instanceRef) reference to model elements out of the context of target-configuration to the model element that sparked the existence of the owner of the reference.  This reference does not carry semantics and is meant for documentation purposes only.  <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	<a href="#">ARObject</a> , <a href="#">ApmcAbstractReferenceValue</a> , <a href="#">ApmcAbstractValue</a> , <a href="#">ApmcContainerElementValue</a> , <a href="#">ApmcIndexableValue</a> , <a href="#">ApmcReferenceValue</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Aggregated by</b>	<a href="#">ApmcContainerValue.element</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
value	<a href="#">Referrable</a>	0..1	ref	This reference identifies the target model element that exists outside the target-configuration, i.e. in the design model.  <b>Stereotypes:</b> atpUriDef <b>Tags:</b> atp.Status=candidate

**Table 4.24: ApmcUpstreamDocForeignReferenceValue**

### [AP\_TPS\_APMC\_01032] Semantics of [ApmcAbstractInstanceReferenceValue](#)

Status: DRAFT

[Abstract meta-class [ApmcAbstractInstanceReferenceValue](#) is used as a base class for the creation of a **structured** reference to an element of the M2 design model.

In other words, all subclasses of [ApmcAbstractInstanceReferenceValue](#) are used to establish a reference from the configuration model to the design model on the “value side”.]

The modeling of [ApmcAbstractInstanceReferenceValue](#) and its sub-classes is sketched in Figure 4.6.

<b>Class</b>	<b>ApmcAbstractInstanceReferenceValue</b> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::ValueSide			
<b>Note</b>	This abstract meta-class represents the ability to establish a instanceRef (as opposed to flat) reference to model elements out of the context of target-configuration.  <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	<a href="#">ARObject</a> , <a href="#">ApmcAbstractValue</a> , <a href="#">ApmcContainerElementValue</a> , <a href="#">ApmcIndexableValue</a> , <a href="#">ApmcReferenceValue</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Subclasses</b>	<a href="#">ApmcInstanceReferenceValue</a> , <a href="#">ApmcUpstreamDocInstanceReferenceValue</a> , <a href="#">ApmcUriInstanceReferenceValue</a>			
<b>Aggregated by</b>	<a href="#">ApmcContainerValue.element</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

**Table 4.25: ApmcAbstractInstanceReferenceValue**

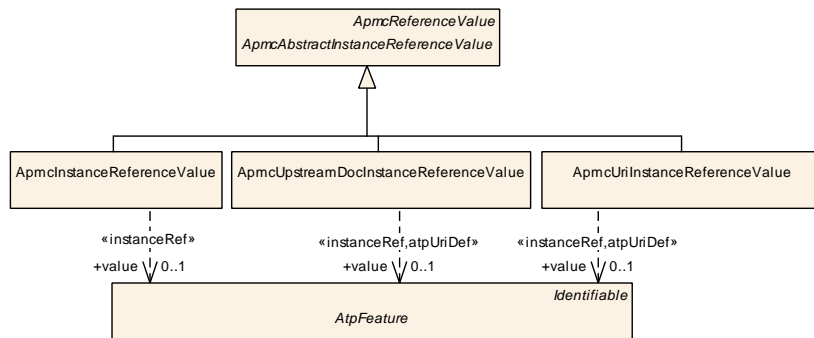


Figure 4.6: Overview of the manifest instance reference values

### [AP\_TPS\_APMC\_01033] Semantics of meta-class [ApmcInstanceReferenceValue](#)

Status: DRAFT

[Meta-class [ApmcInstanceReferenceValue](#) provides the ability to establish a **structured** reference to an [AtpFeature](#) of the M2 model by means of the attribute [value](#).

The target of this reference is required to exist in the scope of the M2 model.]

<b>Class</b>	<b>ApmcInstanceReferenceValue</b>			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::ValueSide			
<b>Note</b>	This meta-class represents the ability to establish a instanceRef (as opposed to flat) reference to model elements out of the context of target-configuration, i.e. in the design model. <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	<a href="#">ARObject</a> , <a href="#">ApmcAbstractInstanceReferenceValue</a> , <a href="#">ApmcAbstractValue</a> , <a href="#">ApmcContainerElementValue</a> , <a href="#">ApmcIndexableValue</a> , <a href="#">ApmcReferenceValue</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Aggregated by</b>	<a href="#">ApmcContainerValue</a> .element			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
value	<a href="#">AtpFeature</a>	0..1	iref	This instanceRef identifies an AtpFeature outside of the target-configuration. <b>Tags:</b> atp.Status=candidate <b>InstanceRef implemented by:</b> <a href="#">AnyInstanceRef</a>

Table 4.26: ApmcInstanceReferenceValue

### [AP\_TPS\_APMC\_01034] Semantics of meta-class [ApmcUriInstanceReferenceValue](#)

Status: DRAFT

[Meta-class [ApmcUriInstanceReferenceValue](#) provides the ability to establish a **structured** reference to an [AtpFeature](#) of the M2 model by means of the attribute [value](#).

The target of this reference is **not** required to exist in the scope of the M2 model. If the target does not exist, then the existence of the reference is taken as an identifying

characteristic of the [ApmcContainerValue](#) that owns the [ApmcUriInstanceReferenceValue](#).]

<b>Class</b>	<b>ApmcUriInstanceReferenceValue</b>			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::ValueSide			
<b>Note</b>	This meta-class represents the ability to establish a instanceRef (as opposed to flat) reference to model elements out of the context of target-configuration, i.e. in the design model.  The referenced model element does not need to exist in the current model scope. The reference itself carries a semantics of being an identifying factor, even without the target element to exist.  <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	<a href="#">ARObject</a> , <a href="#">ApmcAbstractInstanceReferenceValue</a> , <a href="#">ApmcAbstractValue</a> , <a href="#">ApmcContainerElementValue</a> , <a href="#">ApmcIndexableValue</a> , <a href="#">ApmcReferenceValue</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Aggregated by</b>	<a href="#">ApmcContainerValue</a> .element			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
value	<a href="#">AtpFeature</a>	0..1	iref	This instanceRef identifies an AtpFeature outside of the target-configuration that sparked the existence of the owner of the reference.  The referenced model element does not need to exist in the current model scope. The reference itself carries a semantics of being an identifying factor, even without the target element to exist.  <b>Stereotypes:</b> atpUriDef <b>Tags:</b> atp.Status=candidate <b>InstanceRef implemented by:</b> <a href="#">AnyInstanceRef</a>

**Table 4.27: ApmcUriInstanceReferenceValue**

### [AP\_TPS\_APMC\_01035] Semantics of meta-class [ApmcUpstreamDocInstanceReferenceValue](#)

*Status:* DRAFT

[The purpose of the [ApmcUpstreamDocInstanceReferenceValue](#) is to establish a **structured** reference to a **model element in the design domain from which the enclosing container was created by derivation**.

The existence of [ApmcUpstreamDocInstanceReferenceValue](#) is considered as pure meta-data.

In contrast to [ApmcInstanceReferenceDef](#) and [ApmcUriInstanceReferenceDef](#), the [ApmcUpstreamDocInstanceReferenceValue](#) does not carry relevant model semantics and serves only as a protocol of how the enclosing container came into existence.]

<b>Class</b>	<b>ApmcUpstreamDocInstanceReferenceValue</b>			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::AdaptivePlatformMachineConfiguration::ValueSide			
<b>Note</b>	This meta-class represents the ability to establish a instanceRef (as opposed to flat) reference to model elements out of the context of target-configuration to the model element that sparked the existence of the owner of the reference. This reference does not carry semantics and is meant for documentation purposes only. <b>Tags:</b> atp.Status=candidate			
<b>Base</b>	<a href="#">ARObject</a> , <a href="#">ApmcAbstractInstanceReferenceValue</a> , <a href="#">ApmcAbstractValue</a> , <a href="#">ApmcContainerElementValue</a> , <a href="#">ApmcIndexableValue</a> , <a href="#">ApmcReferenceValue</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a>			
<b>Aggregated by</b>	<a href="#">ApmcContainerValue.element</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
value	<a href="#">AtpFeature</a>	0..1	iref	This instanceRef identifies an AtpFeature outside of the target-configuration that sparked the existence of the owner of the reference. <b>Stereotypes:</b> atpUriDef <b>Tags:</b> atp.Status=candidate <b>InstanceRef implemented by:</b> <a href="#">AnyInstanceRef</a>

**Table 4.28: ApmcUpstreamDocInstanceReferenceValue**

## A Reference Material

### A.1 Terms and Abbreviations

The main list of terms and abbreviations are defined in [4]. The following table contains the list of terms and abbreviations used in the scope of this document which are not already defined in [4] along with the spelled-out meaning of each of the abbreviations.

Abbreviation	Meaning
APMC	Adaptive Platform Machine Configuration
ARXML	AUTOSAR XML
DM	Diagnostic Manager
ID	Identifier
ISO	International Standardization Organization
JSON	JavaScript Object Notation
NM	Network Management
NV	Non-Volatile
PHM	Platform Health Management
POSIX	Portable Operating System Interface
ROM	Read-Only Memory
SD	Service Discovery
SDG	Special Data Group
SOME/IP	Scalable service-Oriented MiddlewarE over IP
SOVD	Service-Oriented Vehicle Diagnostics
SWC	Software Component
TLV	Tag Length Value
UDS	Unified Diagnostic Services
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UUID	Universally Unique Identifier
VSA	Variable Size Array

**Table A.1: Terms and Abbreviations used in the scope of this Document**

### A.2 Imposition Times of Constraints

The constraints formulated in this document have different *actual* imposition times which denote the steps in the workflow when the respective constraint has to be imposed.



The imposition times that are considered applicable in the scope of this document<sup>1</sup> are listed in Table A.2.

Please note that the imposition times are intentionally rendered as technical terms such that it is possible to link back from each constraint to the definition of the affected imposition time in Table A.2.

This document has been created to apply primarily for the *AUTOSAR adaptive platform* and therefore the discussed imposition times also apply exclusively to the *AUTOSAR adaptive platform*.

Imposition Time	Description
<b>at the time when the configuration of the functional cluster is finished</b>	This imposition time denotes the step in the workflow, where the configuration of a functional cluster is considered complete such that the installation on a target platform can be started.

**Table A.2: Imposition Times considered in the scope of this document**

---

<sup>1</sup>Different imposition times may be defined in the context of other AUTOSAR standard documents

## **B Examples**

There are no examples in this release.

## C History of Constraints and Specification Items

Please note that the lists in this chapter also include constraints and specification items that have been removed from the specification in a later version. These constraints and specification items do not appear as hyperlinks in the document.

### C.1 Constraint and Specification Item History of this Document according to AUTOSAR R24-11

#### C.1.1 Added Specification Items in R24-11

Number	Heading
[AP_TPS_APMC_01000]	Abstract relation between the "value side" and the "definition side"
[AP_TPS_APMC_01001]	Root element on the "definition side"
[AP_TPS_APMC_01002]	Semantics of <a href="#">ApmcDefinitionCollection</a>
[AP_TPS_APMC_01003]	Specification of element multiplicity on the "definition side"
[AP_TPS_APMC_01004]	Semantics of attribute <a href="#">ApmcDefinitionElement.elementCreation</a>
[AP_TPS_APMC_01005]	Semantics of attribute <a href="#">ApmcConfigurationElementDef.origin</a>
[AP_TPS_APMC_01006]	Grouping of configuration items on the level of a container
[AP_TPS_APMC_01007]	Semantics of a choice container
[AP_TPS_APMC_01008]	Definition of configuration parameter
[AP_TPS_APMC_01009]	Definition of reference
[AP_TPS_APMC_01010]	Root element on the "value side"
[AP_TPS_APMC_01011]	Definition of an enumeration-style parameter
[AP_TPS_APMC_01012]	Definition of default value for enumeration-style parameter
[AP_TPS_APMC_01013]	Semantics of attribute <a href="#">ApmcConfigurationElementDef.requiresIndex</a>
[AP_TPS_APMC_01014]	Semantics of meta-class <a href="#">ApmcContainerReferenceDef</a>
[AP_TPS_APMC_01015]	Semantics of reference <a href="#">ApmcContainerReferenceDef.destination</a>
[AP_TPS_APMC_01016]	Semantics of meta-class <a href="#">ApmcChoiceContainerReferenceDef</a>
[AP_TPS_APMC_01017]	Semantics of attribute <a href="#">ApmcAbstractForeignReferenceDef.destinationType</a>
[AP_TPS_APMC_01018]	Semantics of <a href="#">ApmcForeignReferenceDef</a>
[AP_TPS_APMC_01019]	Semantics of meta-class <a href="#">ApmcUriForeignReferenceDef</a>
[AP_TPS_APMC_01020]	Semantics of meta-class <a href="#">ApmcUpstreamDocForeignReferenceDef</a>
[AP_TPS_APMC_01021]	Semantics of attribute <a href="#">ApmcAbstractInstanceReferenceDef.destinationContext</a>





Number	Heading
[AP_TPS_APMC_01022]	Semantics of attribute <code>ApmcAbstractInstanceReferenceDef.destinationType</code>
[AP_TPS_APMC_01023]	Semantics of <code>ApmcInstanceReferenceDef</code>
[AP_TPS_APMC_01024]	Semantics of meta-class <code>ApmcUriInstanceReferenceDef</code>
[AP_TPS_APMC_01025]	Semantics of meta-class <code>ApmcUpstreamDocInstanceReferenceValue</code>
[AP_TPS_APMC_01026]	Semantics of attribute <code>ApmcIndexableValue.index</code>
[AP_TPS_APMC_01027]	Gaps in the value range of an ordered collection
[AP_TPS_APMC_01028]	Semantics of <code>ApmcAbstractReferenceValue</code>
[AP_TPS_APMC_01029]	Semantics of meta-class <code>ApmcForeignReferenceValue</code>
[AP_TPS_APMC_01030]	Semantics of meta-class <code>ApmcUriForeignReferenceValue</code>
[AP_TPS_APMC_01031]	Semantics of meta-class <code>ApmcUpstreamDocForeignReferenceValue</code>
[AP_TPS_APMC_01032]	Semantics of <code>ApmcAbstractInstanceReferenceValue</code>
[AP_TPS_APMC_01033]	Semantics of meta-class <code>ApmcInstanceReferenceValue</code>
[AP_TPS_APMC_01034]	Semantics of meta-class <code>ApmcUriInstanceReferenceValue</code>
[AP_TPS_APMC_01035]	Semantics of meta-class <code>ApmcUpstreamDocInstanceReferenceValue</code>
[AP_TPS_APMC_01036]	Semantics of <code>ApmcValueCollection</code>
[AP_TPS_APMC_01037]	Semantics of meta-class <code>ApmcContainerReferenceValue</code>
[AP_TPS_APMC_01038]	<code>ApmcFunctionalClusterDef</code> categories

**Table C.1: Added Specification Items in R24-11**

### C.1.2 Changed Specification Items in R24-11

none

### C.1.3 Deleted Specification Items in R24-11

none

### C.1.4 Added Constraints in R24-11

Number	Heading
[constr_10579]	Existence of <a href="#">definition</a>
[constr_10580]	Restriction regarding the value of <a href="#">ApmcDefinitionElement.lowerMultiplicity</a> vs. <a href="#">ApmcDefinitionElement.lowerMultiplicity</a>
[constr_10581]	Existence of reference in the role <a href="#">ApmcFunctionalClusterValue.definition</a>
[constr_10582]	Existence of reference in the role <a href="#">ApmcContainerValue.definition</a>
[constr_10583]	Existence of reference in the role <a href="#">ApmcContainerElementValue.definition</a>
[constr_10584]	Allowed <a href="#">value</a> for a <a href="#">ApmcNumericalParamValue</a> that implements an boolean-style parameter
[constr_10585]	Allowed <a href="#">value</a> for a <a href="#">ApmcTextualParamValue</a> that implements an enumeration-style parameter
[constr_9322]	<a href="#">ApmcFunctionalClusterDef</a> category restriction

**Table C.2: Added Constraints in R24-11**

### C.1.5 Changed Constraints in R24-11

none

### C.1.6 Deleted Constraints in R24-11

none

## D Mentioned Class Tables

For the sake of completeness, this chapter contains a set of class tables representing meta-classes mentioned in the context of this document but which are not contained directly in the scope of describing specific meta-model semantics.

<b>Class</b>	<b>AUTOSAR</b>			
<b>Package</b>	M2::AUTOSARTemplates::AutosarTopLevelStructure			
<b>Note</b>	Root element of an AUTOSAR description, also the root element in corresponding XML documents. <b>Tags:</b> xml.globalElement=true			
<b>Base</b>	<i>ARObject</i>			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
adminData	AdminData	0..1	aggr	This represents the administrative data of an Autosar file. <b>Stereotypes:</b> atpSplitable <b>Tags:</b> atp.Splitkey=adminData xml.sequenceOffset=10
arPackage	ARPackage	*	aggr	This is the top level package in an AUTOSAR model. <b>Stereotypes:</b> atpSplitable; atpVariation <b>Tags:</b> atp.Splitkey=arPackage.shortName, arPackage.variationPoint.shortLabel vh.latestBindingTime=blueprintDerivationTime xml.sequenceOffset=30
fileInfo Comment	FileInfoComment	0..1	aggr	This represents a possibility to provide a structured comment in an AUTOSAR file. <b>Stereotypes:</b> atpStructuredComment <b>Tags:</b> xml.roleElement=true xml.sequenceOffset=-10 xml.typeElement=false
introduction	DocumentationBlock	0..1	aggr	This represents an introduction on the Autosar file. It is intended for example to represent disclaimers and legal notes. <b>Tags:</b> xml.sequenceOffset=20

**Table D.1: AUTOSAR**

<b>Class</b>	<b>AnyInstanceRef</b>			
<b>Package</b>	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::AnyInstanceRef			
<b>Note</b>	Describes a reference to any instance in an AUTOSAR model. This is the most generic form of an instance ref. Refer to the superclass notes for more details.			
<b>Base</b>	<i>ARObject, AtpInstanceRef</i>			
<b>Aggregated by</b>	<a href="#">ApmcInstanceReferenceValue.value</a> , <a href="#">ApmcUpstreamDocInstanceReferenceValue.value</a> , <a href="#">ApmcUriInstanceReferenceValue.value</a> , <a href="#">Collection.collectedInstance</a> , <a href="#">Collection.sourceInstance</a> , <a href="#">DocumentationContext.feature</a> , <a href="#">EcuInstanceReferenceValue.value</a> , <a href="#">FlatInstanceDescriptor.ecuExtractReference</a> , <a href="#">FlatInstanceDescriptor.upstreamReference</a> , <a href="#">RptContainer.byPassPoint</a> , <a href="#">RptHook.rptArHook</a> , <a href="#">SecurityEventReportInstanceValue.object</a> , <a href="#">ViewMap.firstElementInstance</a> , <a href="#">ViewMap.secondElementInstance</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
base	AtpClassifier	1	ref	This is the base from which navigation path begins. <b>Stereotypes:</b> atpDerived





Class	AnyInstanceRef			
contextElement (ordered)	<a href="#">AtpFeature</a>	*	ref	This is one step in the navigation path specified by the instance ref.
target	<a href="#">AtpFeature</a>	1	ref	This is the target of the instance ref.

**Table D.2: AnyInstanceRef**

Class	AtpFeature (abstract)			
Package	M2::AUTOSARTemplates::GenericStructure::AbstractStructure			
Note	Features are properties via which a classifier classifies instances. Or: a classifier has features and every M0 instance of it will have those features.			
Base	<i>ARObject</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
Subclasses	<i>AtpPrototype</i> , <i>AtpStructureElement</i> , <i>PhmCheckpoint</i>			
Aggregated by	<i>AtpClassifier</i> .atpFeature			
Attribute	Type	Mult.	Kind	Note
–	–	–	–	–

**Table D.3: AtpFeature**

Class	EcuInstance			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology			
Note	ECUInstances are used to define the ECUs used in the topology. The type of the ECU is defined by a reference to an ECU specified with the ECU resource description.  <b>Tags:</b> atp.recommendedPackage=EcuInstances			
Base	<i>ARObject</i> , <i>CollectableElement</i> , <i>FibexElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>Referrable</i>			
Aggregated by	ARPackage.element			
Attribute	Type	Mult.	Kind	Note
associated Consumed Provided ServiceInstance Group	ConsumedProvidedServiceInstanceGroup	*	ref	With this reference it is possible to identify which ConsumedProvidedServiceInstanceGroups are applicable for which ECUInstance.  <b>Stereotypes:</b> atpSplittable; atpVariation <b>Tags:</b> atp.Splitkey=associatedConsumedProvidedServiceInstanceGroup.consumedProvidedServiceInstanceGroup, associatedConsumedProvidedServiceInstanceGroup.variationPoint.shortLabel vh.latestBindingTime=postBuild
associatedPdur IPduGroup	PdurIPduGroup	*	ref	With this reference it is possible to identify which Pdur IPdu Groups are applicable for which Communication Connector/ ECU.
channel Synchronous Wakeup	Boolean	0..1	attr	If this parameter is available and set to true, then all available channels will be woken up as soon as at least one channel wakeup occurs. If PNCs are configured, then all PNCs will be requested upon a channel wakeup.
clientIdRange	ClientIdRange	0..1	aggr	Restriction of the Client Identifier for this Ecu to an allowed range of numerical values. The Client Identifier of the transaction handle is generated by the client RTE for inter-Ecu Client/Server communication.





Class	EculInstance			
commController	Communication Controller	*	aggr	CommunicationControllers of the ECU. <b>Stereotypes:</b> atpSplitable; atpVariation <b>Tags:</b> atp.Splitkey=commController.shortName, commController.variationPoint.shortLabel vh.latestBindingTime=postBuild
connector	Communication Connector	*	aggr	All channels controlled by a single controller. <b>Stereotypes:</b> atpSplitable; atpVariation <b>Tags:</b> atp.Splitkey=connector.shortName, connector.variationPoint.shortLabel vh.latestBindingTime=postBuild
firewallRule	StateDependentFirewall	*	ref	Firewall rules defined in the context of an EculInstance. <b>Tags:</b> atp.Status=candidate
pncNmRequest	Boolean	0..1	attr	Defines if this EculInstance shall request Nm on all its PhysicalChannels which have Nm variant set to FULL each time a PNC is requested.
pncPrepare SleepTimer	TimeValue	0..1	attr	Time in seconds the PNC state machine shall wait in PNC_PREPARE_SLEEP.
pnc Synchronous Wakeup	Boolean	0..1	attr	If this parameter is available and set to true then all available PNCs will be woken up as soon as a channel wakeup occurs. This is ensured by adding all PNCs to all channel wakeup sources during upstream mapping.
pnResetTime	TimeValue	0..1	attr	Specifies the runtime of the reset timer in seconds. This reset time is valid for the reset of PN requests in the EIRA and in the ERA.

**Table D.4: EculInstance**

<b>Class</b>	<b>Identifiable</b> (abstract)
<b>Package</b>	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::Identifiable
<b>Note</b>	Instances of this class can be referred to by their identifier (within the namespace borders). In addition to this, Identifiables are objects which contribute significantly to the overall structure of an AUTOSAR description. In particular, Identifiables might contain Identifiables.
<b>Base</b>	<i>ARObject</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>
<b>Subclasses</b>	<i>ARPackage</i> , <i>AbstractDolpLogicAddressProps</i> , <i>AbstractEvent</i> , <i>AbstractFunctionalClusterDesign</i> , <i>AbstractImplementationDataTypeElement</i> , <i>AbstractSecurityEventFilter</i> , <i>AbstractSecurityIdsmInstanceFilter</i> , <i>AbstractServiceInstance</i> , <i>AbstractSignalBasedToISignalTriggeringMapping</i> , <i>AdaptiveSwcInternalBehavior</i> , <i>ApApplicationEndpoint</i> , <i>ApmcAbstractDefinition</i> , <i>ApmcConfigurationElementDef</i> , <i>ApmcContainerElementValue</i> , <i>ApmcContainerValue</i> , <i>ApmcEnumerationLiteralDef</i> , <i>ApplicationEndpoint</i> , <i>ApplicationError</i> , <i>AppliedStandard</i> , <i>ArtifactChecksum</i> , <i>ArtifactLocator</i> , <i>AtpBlueprint</i> , <i>AtpBlueprintable</i> , <i>AtpClassifier</i> , <i>AtpFeature</i> , <i>AutosarOperationArgumentInstance</i> , <i>AutosarVariableInstance</i> , <i>BuildActionEntity</i> , <i>BuildActionEnvironment</i> , <i>Chapter</i> , <i>CheckpointTransition</i> , <i>ClassContentConditional</i> , <i>ClientIdDefinition</i> , <i>ClientServerOperation</i> , <i>Code</i> , <i>CollectableElement</i> , <i>ComManagementMapping</i> , <i>CommConnectorPort</i> , <i>CommunicationConnector</i> , <i>CommunicationController</i> , <i>Compiler</i> , <i>ConsistencyNeeds</i> , <i>ConsumedEventGroup</i> , <i>CouplingPort</i> , <i>CouplingPortAbstractShaper</i> , <i>CouplingPortStructuralElement</i> , <i>CryptoCertificate</i> , <i>CryptoKeySlot</i> , <i>CryptoKeySlotDesign</i> , <i>CryptoKeySlotUsageDesign</i> , <i>CryptoProvider</i> , <i>CryptoServiceMapping</i> , <i>DataPrototypeGroup</i> , <i>DataPrototypeTransformationPropsIdent</i> , <i>DataTransformation</i> , <i>DdsCpDomain</i> , <i>DdsCpPartition</i> , <i>DdsCpQosProfile</i> , <i>DdsCpTopic</i> , <i>DdsDomainRange</i> , <i>DependencyOnArtifact</i> , <i>DiagEventDebounceAlgorithm</i> , <i>DiagnosticAuthTransmitCertificateEvaluation</i> , <i>DiagnosticConnectedIndicator</i> , <i>DiagnosticDataElement</i> , <i>DiagnosticDebounceAlgorithmProps</i> , <i>DiagnosticFunctionInhibitSource</i> , <i>DiagnosticParameterElement</i> , <i>DiagnosticRoutineSubfunction</i> , <i>DiagnosticSovdMethodPrimitive</i> , <i>DltApplication</i> , <i>DltArgument</i> , <i>DltMessage</i> , <i>DolpInterface</i> , <i>DolpLogicAddress</i> , <i>DolpLogicalAddress</i> , <i>DolpNetworkConfigurationDesign</i> , <i>DolpRoutingActivation</i> , <i>E2EProfileConfiguration</i> , <i>End2EndEventProtectionProps</i> , <i>End2EndMethodProtectionProps</i> , <i>EndToEndProtection</i> , <i>Ethernet</i>







Class	Identifiable (abstract)			
	△ WakeupSleepOnDataLineConfig, EventHandler, EventMapping, ExclusiveArea, ExecutableEntity, ExecutionTime, FMAttributeDef, FMFeatureMapAssertion, FMFeatureMapCondition, FMFeatureMapElement, FMFeatureRelation, FMFeatureRestriction, FMFeatureSelection, FieldMapping, FireAndForgetMethodMapping, FlexrayArTpNode, FlexrayTpPduPool, FrameTriggering, GeneralParameter, GlobalSupervision, GlobalTimeGateway, GlobalTimeMaster, GlobalTimeSlave, HealthChannel, HeapUsage, HwAttributeDef, HwAttributeLiteralDef, HwPin, HwPinGroup, IEEE1722TpActBus, IEEE1722TpActBusPart, IPsecRule, IPv6ExtHeaderFilterList, ISignalToIPduMapping, ISignalTriggering, IdentCaption, ImpositionTime, InternalTriggeringPoint, Keyword, LifeCycleState, Linker, MacAddressVlanMembership, MacMulticastGroup, MacSecKeyParticipant, McDataInstance, MemorySection, MemoryUsage, MethodMapping, ModeDeclaration, ModeDeclarationMapping, ModeSwitchPoint, NetworkEndpoint, NmCluster, NmNode, PackageableElement, ParameterAccess, PduActivationRoutingGroup, PduToFrameMapping, PduTriggering, PerInstanceMemory, PersistencyDeploymentElement, PersistencyInterfaceElement, PhmSupervision, PhysicalChannel, PortGroup, PortInterfaceMapping, ProcessToMachineMapping, Processor, ProcessorCore, PskIdentityToKeySlotMapping, ResourceConsumption, ResourceGroup, RootSwClusterDesignComponentPrototype, RootSwComponentPrototype, RootSwCompositionPrototype, RptComponent, RptContainer, RptExecutableEntity, RptExecutableEntityEvent, RptExecutionContext, RptProfile, RptServicePoint, RunnableEntityGroup, SdgAttribute, SdgClass, SecOcJobMapping, SecOcJobRequirement, SecureCommunicationAuthenticationProps, SecureCommunicationDeployment, SecureCommunicationFreshnessProps, SecurityEventContextDataElement, SecurityEventContextProps, ServiceEventDeployment, ServiceFieldDeployment, ServiceInterfaceElementSecureComConfig, ServiceMethodDeployment, ServiceNeeds, SignalServiceTranslationEventProps, SignalServiceTranslationProps, SocketAddress, SoftwarePackageStep, SomeipEventGroup, SomeipProvidedEventGroup, SomeipTpChannel, SpecElementReference, StackUsage, StateManagementActionItem, StateManagementActionList, StateManagementStateNotification, StateManagementStateRequest, StaticSocketConnection, StructuredReq, SupervisionCheckpoint, SupervisionMode, SupervisionModeCondition, SwGenericAxisParamType, SwServiceArg, SwcServiceDependency, SwitchAsynchronousTrafficShaperGroupEntry, SystemMapping, TimeBaseResource, TimingClock, TimingClockSyncAccuracy, TimingCondition, TimingConstraint, TimingDescription, TimingExtensionResource, TimingModeInstance, TlsCryptoCipherSuite, TlsCryptoCipherSuiteProps, TlsJobMapping, Topic1, TpAddress, TraceableTable, TraceableText, TracedFailure, TransformationISignalPropsIdent, TransformationProps, TransformationTechnology, Trigger, UcmDescription, UcmRetryStrategy, UcmStep, VariableAccess, VariationPointProxy, VehicleRolloutStep, ViewMap, VlanConfig, WaitPoint			
Attribute	Type	Mult.	Kind	Note
adminData	AdminData	0..1	aggr	This represents the administrative data for the identifiable object.  <b>Stereotypes:</b> atpSplitable <b>Tags:</b> atp.Splitkey=adminData xml.sequenceOffset=-40
annotation	Annotation	*	aggr	Possibility to provide additional notes while defining a model element (e.g. the ECU Configuration Parameter Values). These are not intended as documentation but are mere design notes.  <b>Tags:</b> xml.sequenceOffset=-25
category	CategoryString	0..1	attr	The category is a keyword that specializes the semantics of the Identifiable. It affects the expected existence of attributes and the applicability of constraints.  <b>Tags:</b> xml.sequenceOffset=-50
desc	MultiLanguageOverviewParagraph	0..1	aggr	This represents a general but brief (one paragraph) description what the object in question is about. It is only one paragraph! Desc is intended to be collected into overview tables. This property helps a human reader to identify the object in question.  More elaborate documentation, (in particular how the object is built or used) should go to "introduction".  <b>Tags:</b> xml.sequenceOffset=-60





<b>Class</b>	<b>Identifiable</b> (abstract)			
introduction	DocumentationBlock	0..1	aggr	This represents more information about how the object in question is built or is used. Therefore it is a DocumentationBlock. <b>Tags:</b> xml.sequenceOffset=-30
uuid	String	0..1	attr	The purpose of this attribute is to provide a globally unique identifier for an instance of a meta-class. The values of this attribute should be globally unique strings prefixed by the type of identifier. For example, to include a DCE UUID as defined by The Open Group, the UUID would be preceded by "DCE:". The values of this attribute may be used to support merging of different AUTOSAR models. The form of the UUID (Universally Unique Identifier) is taken from a standard defined by the Open Group (was Open Software Foundation). This standard is widely used, including by Microsoft for COM (GUIDs) and by many companies for DCE, which is based on CORBA. The method for generating these 128-bit IDs is published in the standard and the effectiveness and uniqueness of the IDs is not in practice disputed. If the id namespace is omitted, DCE is assumed. An example is "DCE:2fac1234-31f8-11b4-a222-08002b34c003". The uuid attribute has no semantic meaning for an AUTOSAR model and there is no requirement for AUTOSAR tools to manage the timestamp. <b>Tags:</b> xml.attribute=true

**Table D.5: Identifiable**

<b>Class</b>	<b>Machine</b>			
<b>Package</b>	M2::AUTOSARTemplates::AdaptivePlatform::SubSystemDesign::MachineManifest			
<b>Note</b>	Machine that represents an Adaptive Autosar Software Stack. <b>Tags:</b> atp.recommendedPackage=Machines			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">AtpClassifier</a> , <a href="#">AtpFeature</a> , <a href="#">AtpStructureElement</a> , <a href="#">CollectableElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a> , <a href="#">UploadableDeploymentElement</a> , <a href="#">UploadablePackageElement</a>			
<b>Aggregated by</b>	ARPackage.element, <a href="#">AtpClassifier</a> .atpFeature			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
default Application Timeout	EnterExitTimeout	0..1	aggr	This aggregation defines a default timeout in the context of a given Machine with respect to the launching and termination of applications.
environment Variable	TagWithOptionalValue	*	aggr	This aggregation represents the collection of environment variables that shall be added to the environment defined on the level of the enclosing Machine. <b>Stereotypes:</b> atpSplitable <b>Tags:</b> atp.Splitkey=environmentVariable
machineDesign	MachineDesign	0..1	ref	Reference to the MachineDesign this Machine is implementing.
module Instantiation	AdaptiveModule Instantiation	*	aggr	Configuration of Adaptive Autosar module instances that are running on the machine. <b>Stereotypes:</b> atpSplitable <b>Tags:</b> atp.Splitkey=moduleInstantiation.shortName
processor	Processor	*	aggr	This represents the collection of processors owned by the enclosing machine.





Class	Machine			
secureCommunicationDeployment	SecureCommunicationDeployment	*	aggr	Target-configuration of secure communication protocol configuration settings to crypto module entities.  <b>Stereotypes:</b> atpSplittable <b>Tags:</b> atp.Splitkey=secureCommunicationDeployment.shortName
trustedPlatformExecutableLaunchBehavior	TrustedPlatformExecutableLaunchBehaviorEnum	0..1	attr	This attribute controls the behavior of how authentication affects the ability to launch for each Executable.

**Table D.6: Machine**

Class	PortPrototype (abstract)			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components			
Note	Base class for the ports of an AUTOSAR software component.  The aggregation of PortPrototypes is subject to variability with the purpose to support the conditional existence of ports.			
Base	ARObject, AtpBlueprintable, AtpFeature, AtpPrototype, Identifiable, MultilanguageReferrable, Referrable			
Subclasses	AbstractProvidedPortPrototype, AbstractRequiredPortPrototype			
Aggregated by	AtpClassifier.atpFeature, SwComponentType.port			
Attribute	Type	Mult.	Kind	Note
clientServerAnnotation	ClientServerAnnotation	*	aggr	Annotation of this PortPrototype with respect to client/server communication.
delegatedPortAnnotation	DelegatedPortAnnotation	0..1	aggr	Annotations on this delegated port.
ioHwAbstractionServerAnnotation	IoHwAbstractionServerAnnotation	*	aggr	Annotations on this IO Hardware Abstraction port.
modePortAnnotation	ModePortAnnotation	*	aggr	Annotations on this mode port.
nvDataPortAnnotation	NvDataPortAnnotation	*	aggr	Annotations on this non volatile data port.
parameterPortAnnotation	ParameterPortAnnotation	*	aggr	Annotations on this parameter port.
portPrototypeProps	PortPrototypeProps	0..1	aggr	This attribute allows for the definition of further qualification of the semantics of a PortPrototype.
senderReceiverAnnotation	SenderReceiverAnnotation	*	aggr	Collection of annotations of this ports sender/receiver communication.
triggerPortAnnotation	TriggerPortAnnotation	*	aggr	Annotations on this trigger port.

**Table D.7: PortPrototype**

Class	ProcessDesign
Package	M2::AUTOSARTemplates::AdaptivePlatform::ApplicationDesign::ProcessDesign
Note	This meta-class has the ability to stand in for a Process at the time when the Process does not yet exist. But its future existence already needs to be considered during design phase and for that a dedicated model element is required..  <b>Tags:</b> atp.recommendedPackage=ProcessDesigns
Base	ARElement, ARObject, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable
Aggregated by	ARPackage.element





<b>Class</b>	<b>ProcessDesign</b>			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
executable	Executable	*	ref	Reference to executable that is executed in the process.

**Table D.8: ProcessDesign**

<b>Class</b>	<b>Referrable</b> (abstract)			
<b>Package</b>	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::Identifiable			
<b>Note</b>	Instances of this class can be referred to by their identifier (while adhering to namespace borders).			
<b>Base</b>	ARObject			
<b>Subclasses</b>	<i>AtpDefinition</i> , <i>BswDistinguishedPartition</i> , <i>BswModuleCallPoint</i> , <i>BswModuleClientServerEntry</i> , <i>BswVariableAccess</i> , <i>CouplingPortTrafficClassAssignment</i> , <i>CppImplementationDataTypeContextTarget</i> , <i>DiagnosticEnvModeElement</i> , <i>EthernetPriorityRegeneration</i> , <i>ExclusiveAreaNestingOrder</i> , <i>HwDescriptionEntity</i> , <i>ImplementationProps</i> , <i>ModeTransition</i> , <i>MultilanguageReferrable</i> , <i>NmNetworkHandle</i> , <i>PncMappingIdent</i> , <i>SingleLanguageReferrable</i> , <i>SoConIPdulIdentifier</i> , <i>SocketConnectionBundle</i> , <i>SomeipRequiredEventGroup</i> , <i>TimeSyncServerConfiguration</i> , <i>TpConnectionIdent</i>			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
shortName	Identifier	1	attr	This specifies an identifying shortName for the object. It needs to be unique within its context and is intended for humans but even more for technical reference.  <b>Stereotypes:</b> atpIdentityContributor <b>Tags:</b> xml.enforceMinMultiplicity=true xml.sequenceOffset=-100
shortNameFragment	ShortNameFragment	*	aggr	This specifies how the Referrable.shortName is composed of several shortNameFragments.  <b>Tags:</b> xml.sequenceOffset=-90

**Table D.9: Referrable**