

<b>Document Title</b>	Supplementary material of Workflow Example
<b>Document Owner</b>	AUTOSAR
<b>Document Responsibility</b>	AUTOSAR
<b>Document Identification No</b>	1145

<b>Document Status</b>	published
<b>Part of AUTOSAR Standard</b>	Classic Platform
<b>Part of Standard Release</b>	R25-11

Document Change History			
Date	Release	Changed by	Description
2025-07-22	1.0.0	Bosch / ETAS	<ul style="list-style-type: none"> <li>initial version to collect supplementary information</li> <li>include first review findings of iteration 01</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.

## Table of Contents

1	Introduction	4
1.1	Objectives . . . . .	4
1.2	Scope . . . . .	4
2	Example System	6
2.1	Technical Aspects . . . . .	9
2.2	Architectural Aspects . . . . .	10
3	Related Documentation	15
A	Delivered artefacts	16
B	Mentioned Class Tables	17

# 1 Introduction

This technical report provides additional information to the AUTOSAR Workflow Example and is intended for all those who are concerned with the example.

## 1.1 Objectives

The objectives for the Workflow Example are:

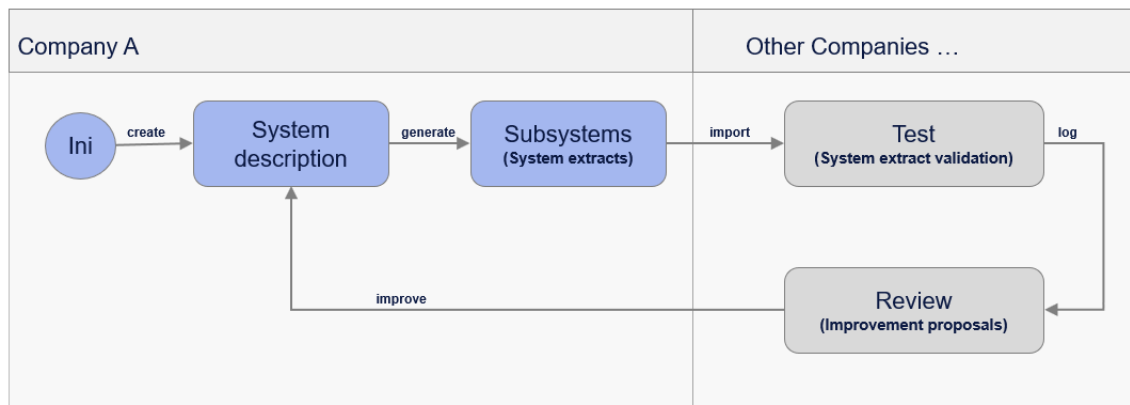
- AUTOSAR is committed to enhance user experience by providing practical guides and examples to ease the entry into the work with AUTOSAR and to show typical applications.
- Since AUTOSAR specifies exchange artifacts, we want to provide practical examples based on these standards that can be processed by as many implementation providers as possible.
- To verify the usability and quality of these exchange artifacts, the processed artifacts must be integrable and systemically functional after deployment to a hardware target.
- This kind of test corresponds to the idea of a plug-fest, in which different providers plug their products installed on a hardware together and provide proof that they can work together.

## 1.2 Scope

The basis of the Workflow Example is an iterative process which shall handle the challenges of AUTOSAR projects, e.g.

- specifications still leave room for interpretation,
- incomplete implementations of the standard,
- errors that unfortunately creep-in from time to time.

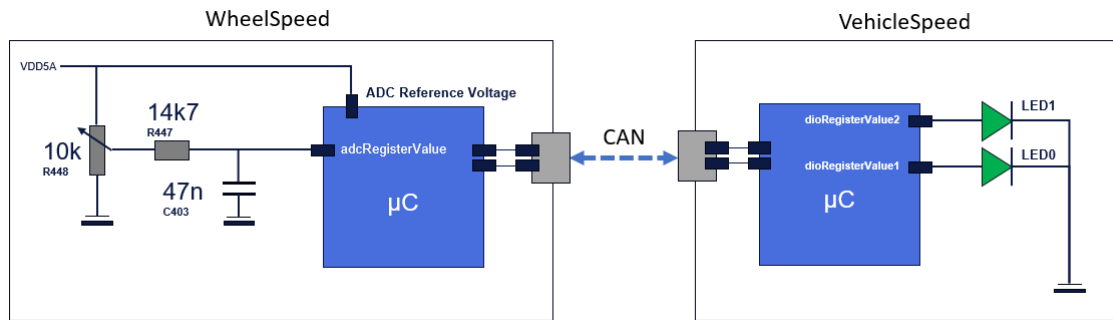
The workflow example provides a system description from which development partners can generate the system extracts for the individual sub-systems in such a way that as many partners as possible can process them smoothly with their tools.



**Figure 1.1: Structure of iterative process**

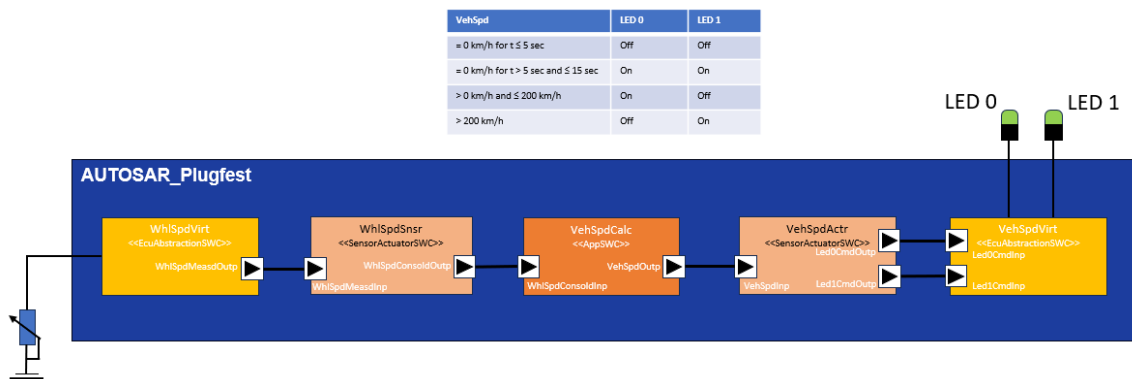
## 2 Example System

The basis function of the system is that a wheel speed which is simulated by a potentiometer, converted by an ADC module (Analog-Digital-Converter) of the WheelSpeed ECU, transmitted via CAN to the VehicleSpeed ECU and the calculated vehicle speed is indicated by two LEDs by a DIO module (Digital-Input-Output).



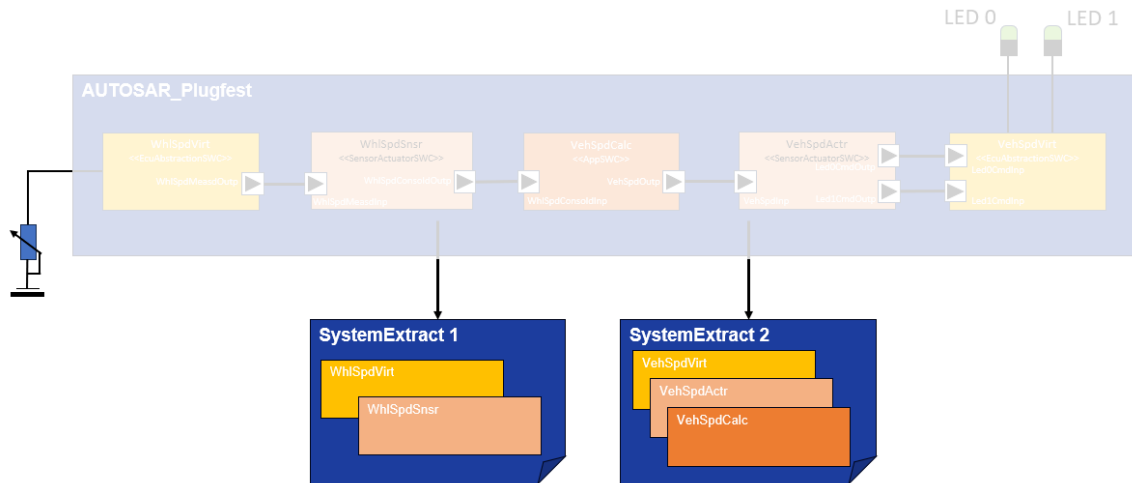
**Figure 2.1: Functional sketch of the system**

Transferred in the AUTOSAR methodology the function is defined by five software components in one system composition. Two [SensorActuatorSwComponentTypes](#) for the interaction with outer world, one [SwComponentType](#) for the internal calculation and two [EcuAbstractionSwComponentType](#) for the abstraction of the hardware. The [System](#) class is used to describe the System Configuration of the complete AUTOSAR system. The category property of [System](#) shall indicate the role of this work product. Here the category `SYSTEM_DESCRIPTION` is used. See [constr\_01003] in [1]. This is typically done by an OEM.



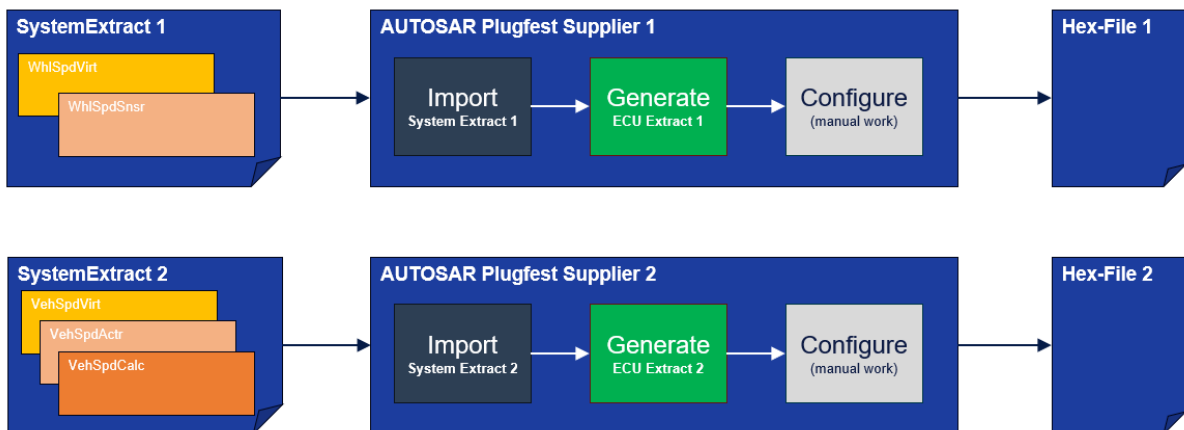
**Figure 2.2: Functional representation in AUTOSAR**

For the Workflow Example the overall system composition is separated in two `SYSTEM_EXTRACTS` assigned each to a separate ECU. These `SYSTEM_EXTRACTS` (SystemExtract 1 and 2) are provided to two different suppliers.



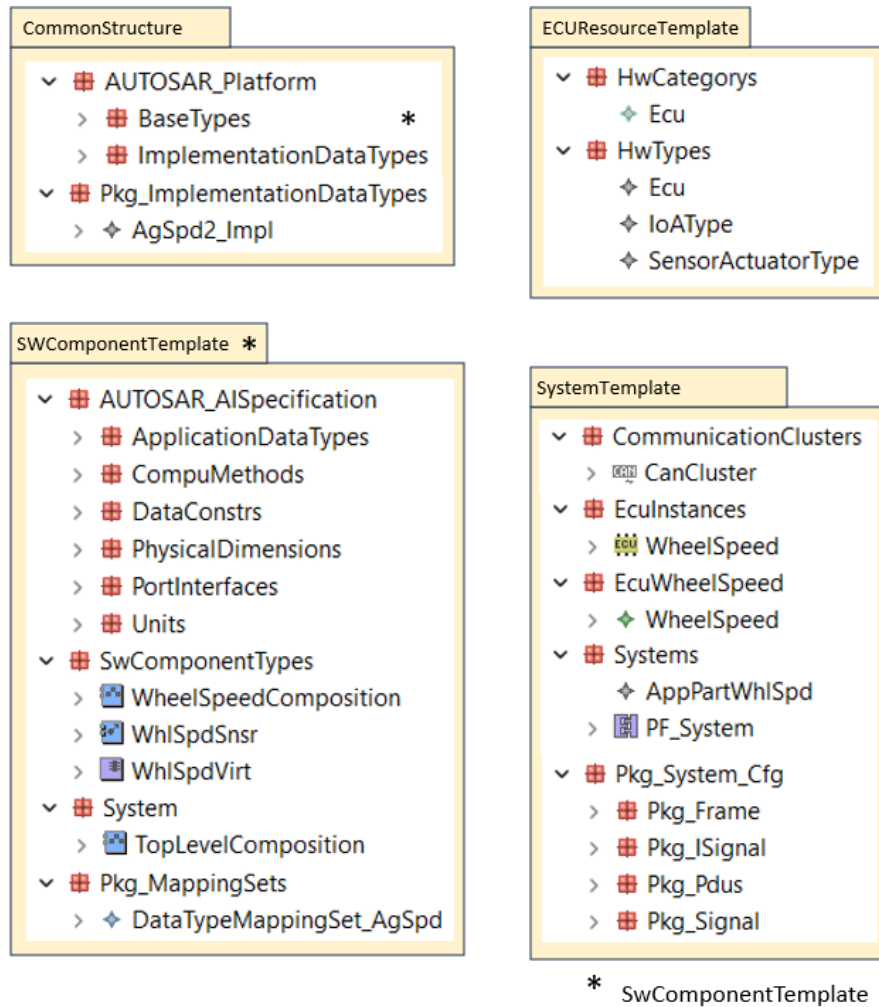
**Figure 2.3: Separation in two System Extracts**

On the supplier side the `SYSTEM_EXTRACT` is imported and all required data are added to generate a sub-system for an ECU which will be later integrated in the complete system by the original requester.



**Figure 2.4: Processing of System Extracts on supplier side**

The provided `SYSTEM_EXTRACT` contains several methodological aspects defined by the different **TemPlate Specifications** (TPS documents).

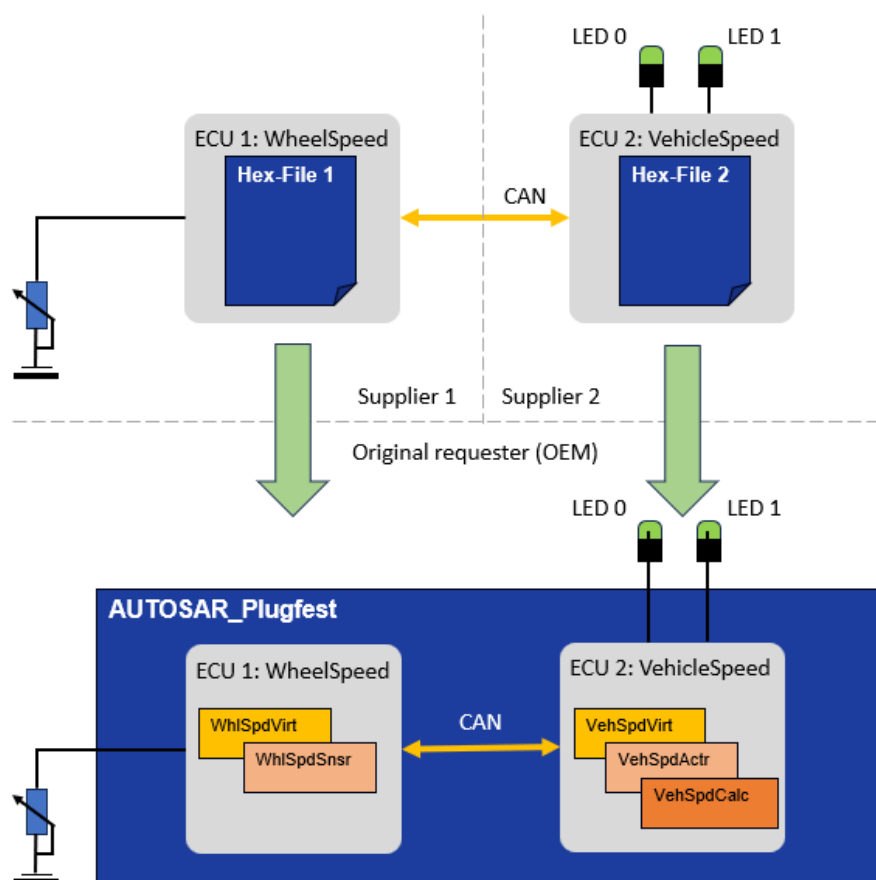


**Figure 2.5: Assignment of System Extract artifacts to dedicated Template Specifications**

The packaging of the artifacts is described in [GenericStructure \[2\]](#). A package may contain an arbitrary number of elements, represented by the abstract [ARElement](#). Note that the aggregation of [ARElement](#) in [ARPackage](#) is subject to variation. Therefore the illustration in 2.5 is only one possible representation. The [CommonStructure](#) is described in [2]. The other packages of the meta-model as the [SystemTemplate \[1\]](#), the [SWComponentTemplate \[3\]](#) and the [ECUResourceTemplate \[4\]](#) and their dependencies to each other are described in [2] - Organization of the Meta-Model.

In the last step, the ECUs provided by the different suppliers, will be integrated in the target system.

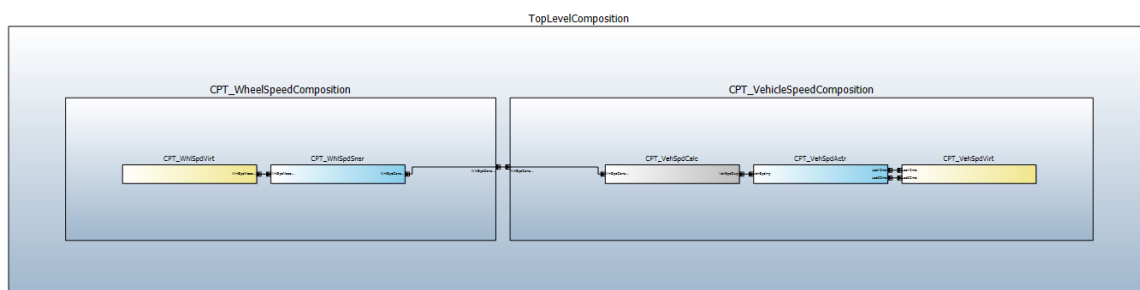




**Figure 2.6: Provided ECUs are integrated in target system**

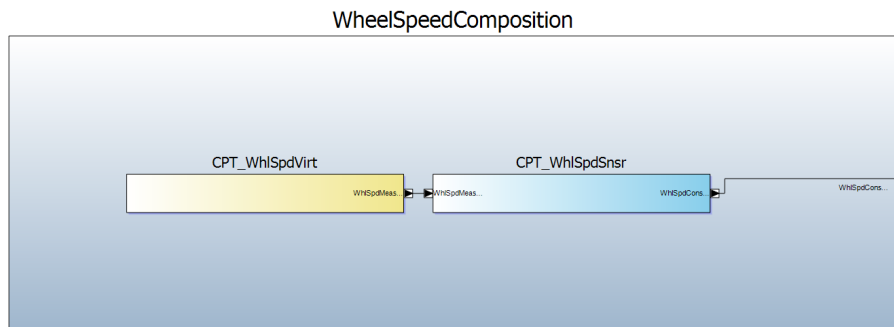
## 2.1 Technical Aspects

The system defined in the `SYSTEM_DESCRIPTION` consists of the "TopLevelComposition" containing two sub-compositions.



**Figure 2.7: TopLevelComposition of designed system**

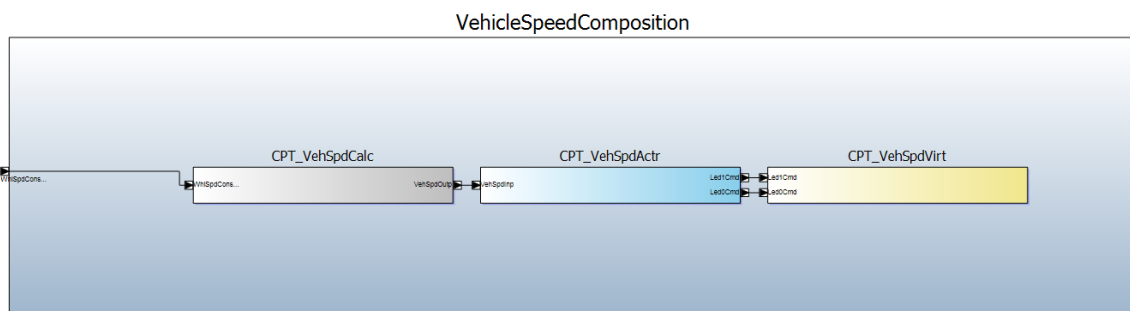
These two sub-compositions are used to generate a `SYSTEM_EXTRACT` for each ECU. The first `SYSTEM_EXTRACT` is dedicated for the ECU "WheelSpeed".



**Figure 2.8: TopLevelComposition of sub-system "WheelSpeed"**

This composition consists of the [EcuAbstractionSwComponentType](#) named "CPT\_WhlSpdVirt" which abstracts the hardware and the [SensorActuatorSwComponentType](#) named "CPT\_WhlSpdSnsr" which handles the sensor signal.

The second `SYSTEM_EXTRACT` is dedicated for the ECU "VehicleSpeed".



**Figure 2.9: TopLevelComposition of sub-system "VehicleSpeed"**

This composition consists of three components: the [ApplicationSwComponentType](#) "CPT\_VehSpdCalc" receives the data from the other ECU and converts the vehicle speed from the wheel speed. The [SensorActuatorSwComponentType](#) "CPT\_VehSpdActr" prepares the actuator signal for each LED based on the calculated vehicle speed which will be finally received by the [EcuAbstractionSwComponentType](#) "CPT\_VehSpdVirt" which encasulates the access to the MCAL drivers for the LED.

## 2.2 Architectural Aspects

Two approaches - top-down and bottom-up - are implemented in this project and will be described in a brief way.

### Hint to delivered artefacts!

The first iteration of this project only contains the artefacts of the top-down approach. See appendix [A](#)

### Top-down Approach

The top-down approach is described in chapter 13.3 SW component inclusion and top level data mapping [1] and also used in this project.

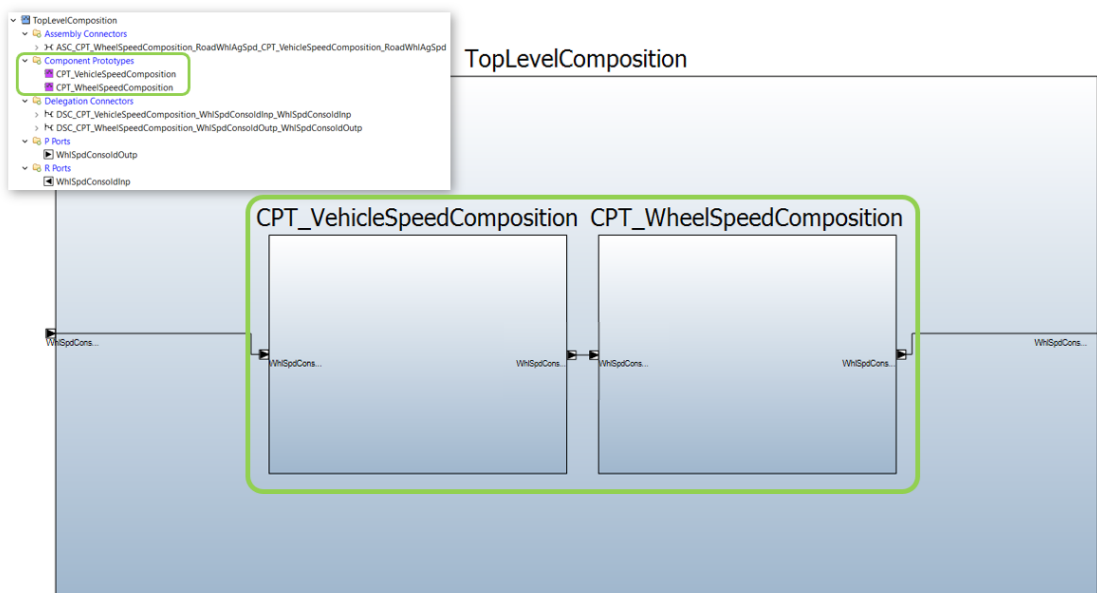
The system designer of the "TopLevelComposition" can design the system in such a way that

- exchangeability on application side and
- decoupling of data mapping between network architecture and application architecture

are achievable. He creates the "TopLevelComposition" based on the [RootSwCompositionPrototype](#) and references the [CompositionSwComponentType](#) as the top-level software composition. The RootSwComposition will usually not contain all atomic software components that are used in a complete VFB (Virtual Function Bus) system.

Instead, the focus is on the interfaces between the components or whole compositions. The system design could even contain only compositions and their ports. In this example, the components are included as well.

The system designer (e.g. OEM) is primarily interested in the required functionality and the interfaces defining the integration of the software compositions or software components into the system. In this project the system designer embeds the two sub-compositions "CPT\_WheelSpeedComposition" and "CPT\_VehicleSpeedComposition" and the required infrastructure.



**Figure 2.10: Design of TopLevelComposition with two sub-compositions (top-down approach)**

An essential aspect of this architectural approach is to decouple the [DataMapping](#) of the network communication from that of the implemented applications. The [PortPro-](#)

types realizing the network communication are defined at the "TopLevelComposition". The corresponding DataMapping, implemented by the subclass Sender-ReceiverToSignalMapping, is also defined referring to the outer PortPrototypes.

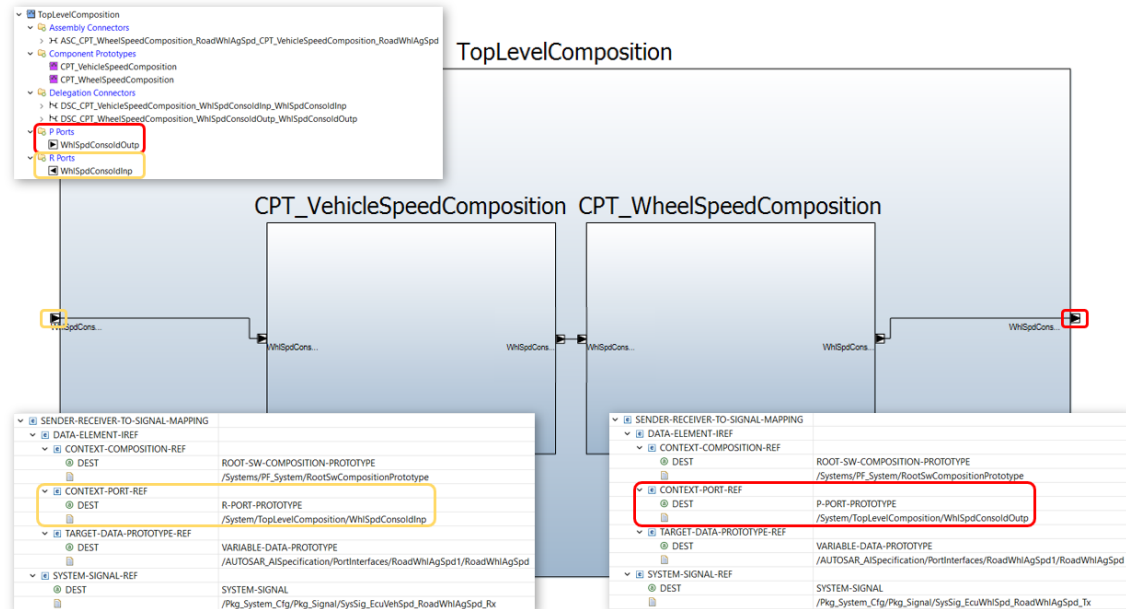


Figure 2.11: Outer PortPrototypes realize network communication

From a VFB based point of view the outer PortPrototypes are delegated to the inner PortPrototypes.

Connection Type	Connect or Name	Component Prototype / Composition	Transmit Port		Connected Component Prototype / Composition	Receive Port		File Name
			Port	Port Interface		Port	Port Interface	
Delegation	DSC_C	CPT_WheelSpeedComposition	WhlSpdConsolidOutp	RoadWhlAgSpd1	TopLevelComposition	WhlSpdConsolidOutp	RoadWhlAgSpd1	PF_SysDesc.arxml
Delegation	DSC_C	TopLevelComposition	WhlSpdConsolidInp	RoadWhlAgSpd1	CPT_VehicleSpeedComposition	WhlSpdConsolidInp	RoadWhlAgSpd1	PF_SysDesc.arxml

Figure 2.12: Delegation of outer PortPrototypes to inner ones

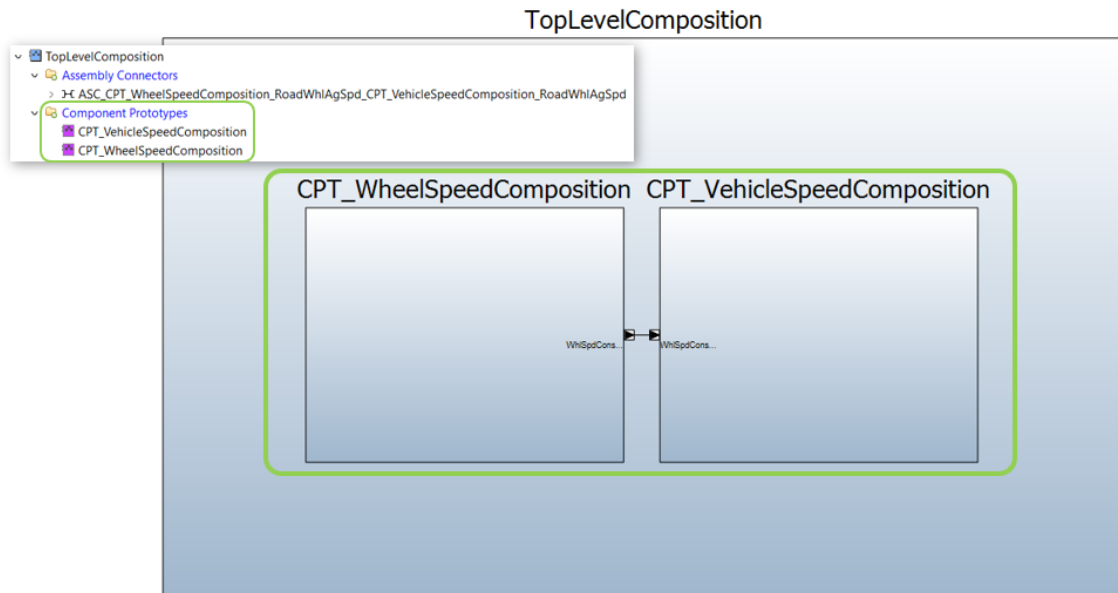
The benefit of the described design approach is that actually the software component architecture inside the root composition can be changed without affecting the DataMappings and thus the network definition. Therefore the supplier of the embedded software compositions or software components can act independently.

## Bottom-Up Approach

The bottom-up approach is described in chapter 13.2 Data Mapping in the System Extract in [1] and also validated as an alternative design approach in the context of the Workflow Example.

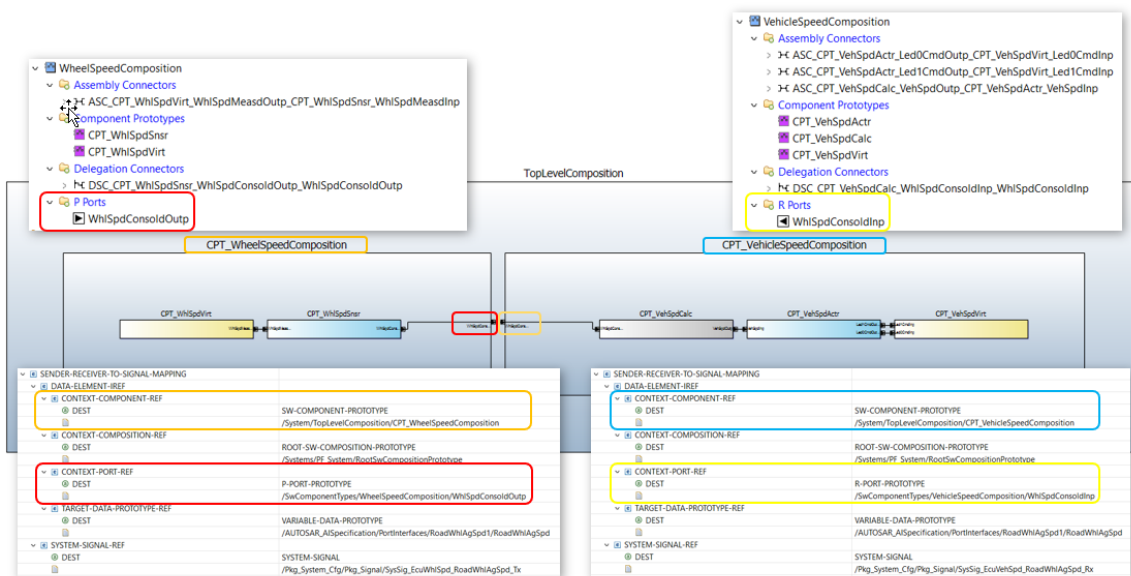
The system design is created bottom-up. In order to assemble applications from AUTOSAR components (AtomicSwComponentType), compositions can be build up hierarchically, until the outermost CompositionSwComponentType forms a kind of top-level composition. This mainly bases on [constr\_3031] in [1], which tells that the unique feature of the complete system description is, that it doesn't have any outside ports,

but all the software components contained in it are connected to each other ([AssemblySwConnector](#)) and fully specified by their [SwComponentTypes](#), [PortPrototypes](#), [PortInterfaces](#), [VariableDataPrototypes](#), [InternalBehavior](#) etc..



**Figure 2.13: Design of TopLevelComposition with two sub-compositions (Bottom-Up Approach)**

In this approach, it is convenient to attach the [DataMapping](#) directly to the software component. The [PortPrototypes](#) are assigned to the [SwComponentTypes](#) of "CPT\_WheelSpeedComposition" and "CPT\_VehicleSpeedComposition".



**Figure 2.14: PortPrototypes are assigned to SwComponentTypes**

Both software compositions are self-contained and connected by [AssemblySwConnector](#) on the next higher composition level, here the "TopLevelComposition", see figure 2.13.

Connection Type	Connector Name	Component Prototype / Composition	Transmit Port		Connected Component Prototype / Composition	Receive Port			File Name
			Port	Port Interface		Port	Port Interface	May Be	
Assembly	ASC_C	CPT_WheelSpeedComposition	WhlSpdConsolidOutput	RoadWhlAgSpd1	CPT_VehicleSpeedComposition	WhlSpdConsolidInp	RoadWhlAgSpd1		PF_SysDesc.arxml

**Figure 2.15: Connection of software compositions by [AssemblySwConnector](#) on next higher composition level**

The benefit of the described design approach is that the designer of an enveloping composition uses the functionalities to be included, in form of software components or compositions, and builds up stepwise the complete system description. This is helpful if the OEM is not able to provide an system description at all.

### 3 Related Documentation

- [1] System Template  
AUTOSAR\_CP\_TPS\_SystemTemplate
- [2] Generic Structure Template  
AUTOSAR\_FO\_TPS\_GenericStructureTemplate
- [3] Software Component Template  
AUTOSAR\_CP\_TPS\_SoftwareComponentTemplate
- [4] Specification of ECU Resource Template  
AUTOSAR\_CP\_TPS\_ECUResourceTemplate

## A Delivered artefacts

The delivery of the first iteration of the Workflow Example (*WorkflowExample.zip*) only includes the artefacts of the top-down approach, see section 2.2:

These are:

- ECU\_SYSTEM\_DESCRIPTION of the Workflow Example project containing all relevant ARXML files regarding to the infrastructure, software components and system configuration. Used and provided by the OEM.
- SYSTEM\_EXTRACT of ECU 1 for the WheelSpeed composition (*WheelSpeed\_\_SystemExtract.arxml*), used by the supplier of ECU 1 for further processing.
- SYSTEM\_EXTRACT of ECU 2 for the VehicleSpeed composition (*VehicleSpeed\_\_SystemExtract.arxml*), used by the supplier of ECU 2 for further processing.
- AUTOSAR\_CP\_TR\_WorkflowExample.pdf, documentation of the Workflow Example, this file.

### Instruction for user of delivered artefacts!

1. Download the *WorkflowExample.zip* from the AUTOSAR Webpage - Classic Platform.
2. Unzip *WorkflowExample.zip*, three folders will be created:
  - EcuSystemDescription,
  - Ecu1SystemExtract and
  - Ecu2SystemExtract.
3. Depending on your role (OEM, supplier) load the corresponding folder in your AUTOSAR tooling and start working. As an OEM work based on the EcuSystemDescription, as a supplier use the System Extracts of ECU1 or ECU2.
4. As supplier continue as illustrated in figure 2.4 and complete the sub-system.
5. As OEM integrate the provided ECUs as illustrated in figure 2.6.
6. Job done.



## B 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>ARElement</b> (abstract)			
<b>Note</b>	An element that can be defined stand-alone, i.e. without being part of another element (except for packages of course).			
<b>Base</b>	<i>ARObject</i> , <i>CollectableElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>Referrable</i>			
<b>Subclasses</b>	AclObjectSet, AclOperation, AclPermission, AclRole, AliasNameSet, ApplicabilityInfoSet, Application Partition, <i>AutosarDataType</i> , <i>BaseType</i> , BlueprintMappingSet, BswEntryRelationshipSet, BswModule Description, BswModuleEntry, BuildActionManifest, CalibrationParameterValueSet, ClientIdDefinitionSet, ClientServerInterfaceToBswModuleEntryBlueprintMapping, Collection, CompuMethod, Consistency NeedsBlueprintSet, ConstantSpecification, ConstantSpecificationMappingSet, CpSoftwareCluster, Cp SoftwareClusterBinaryManifestDescriptor, CpSoftwareClusterMappingSet, CpSoftwareClusterResource Pool, CryptoEllipticCurveProps, CryptoServiceCertificate, CryptoServiceKey, CryptoServicePrimitive, CryptoServiceQueue, CryptoSignatureScheme, DataConstr, DataTransformationSet, DataTypeMapping Set, DdsCpConfig, <i>DiagnosticCommonElement</i> , DiagnosticConnection, DiagnosticContributionSet, Dlt ArgumentPropsSet, DltContext, DltEcu, Documentation, E2EProfileCompatibilityProps, EcucDefinition Collection, EcucDestinationUriDefSet, EcucModuleConfigurationValues, EcucModuleDef, EcucValue Collection, EthIpProps, EthTcpIpCompProps, EthTcpIpProps, EvaluatedVariantSet, FMFeature, FMFeatureMap, FMFeatureModel, FMFeatureSelectionSet, FirewallRule, FlatMap, GeneralPurpose Connection, HwCategory, HwElement, HwType, <i>IEEE1722TpConnection</i> , IPsecConfigProps, IPv6Ext HeaderFilterSet, <i>IdsCommonElement</i> , IdsDesign, <i>Implementation</i> , ImpositionTimeDefinitionGroup, InterpolationRoutineMappingSet, J1939ControllerApplication, KeywordSet, LifeCycleInfoSet, LifeCycle StateDefinitionGroup, LogAndTraceMessageCollectionSet, MacSecGlobalKeyProps, MacSecParticipant Set, McFunction, McGroup, ModeDeclarationGroup, ModeDeclarationMappingSet, OsTaskProxy, PhysicalDimension, PhysicalDimensionMappingSet, <i>PortInterface</i> , PortInterfaceMappingSet, Port PrototypeBlueprint, PostBuildVariantCriterion, PostBuildVariantCriterionValueSet, PredefinedVariant, RapidPrototypingScenario, SdgDef, <i>SecureComProps</i> , SignalServiceTranslationPropsSet, SomeIpSd ClientEventGroupTimingConfig, SomeIpSdClientServiceInstanceConfig, SomeIpSdServerEventGroup TimingConfig, SomeIpSdServerServiceInstanceConfig, SwAddrMethod, SwAxisType, SwComponent MappingConstraints, <i>SwComponentType</i> , SwRecordLayout, SwSystemconst, SwSystemconstantValue Set, SwCbswMapping, <i>System</i> , SystemSignal, SystemSignalGroup, TdCpSoftwareClusterMappingSet, TcpOptionFilterSet, <i>TimingExtension</i> , TlsConnectionGroup, TlvDataIdDefinitionSet, TransformationProps Set, Unit, UnitGroup, <i>UploadablePackageElement</i> , ViewMapSet			
<b>Aggregated by</b>	<a href="#">ARPackage.element</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–

Table B.1: ARElement

<b>Class</b>	<b>ARPackage</b>			
<b>Note</b>	AUTOSAR package, allowing to create top level packages to structure the contained ARElements. ARPackages are open sets. This means that in a file based description system multiple files can be used to partially describe the contents of a package. This is an extended version of MSR's SW-SYSTEM.			
<b>Base</b>	<i>ARObject</i> , <i>AtpBlueprint</i> , <i>AtpBlueprintable</i> , <i>CollectableElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
<b>Aggregated by</b>	<a href="#">ARPackage.arPackage</a> , AUTOSAR.arPackage			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
–	–	–	–	–





Class	ARPackage			
arPackage	<a href="#">ARPackage</a>	*	aggr	This represents a sub package within an ARPackage, thus allowing for an unlimited package hierarchy. <b>Stereotypes:</b> atpSplittable; atpVariation <b>Tags:</b> atp.Splitkey=arPackage.shortName, arPackage.variation Point.shortLabel vh.latestBindingTime=blueprintDerivationTime xml.sequenceOffset=30
element	PackageableElement	*	aggr	Elements that are part of this package <b>Stereotypes:</b> atpSplittable; atpVariation <b>Tags:</b> atp.Splitkey=element.shortName, element.variation Point.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=20
referenceBase	ReferenceBase	*	aggr	This denotes the reference bases for the package. This is the basis for all relative references within the package. The base needs to be selected according to the base attribute within the references. <b>Stereotypes:</b> atpSplittable <b>Tags:</b> atp.Splitkey=referenceBase.shortLabel xml.sequenceOffset=10

**Table B.2: ARPackage**

Class	ApplicationSwComponentType			
<b>Note</b>	The ApplicationSwComponentType is used to represent the application software. <b>Tags:</b> atp.recommendedPackage=SwComponentTypes			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">AtomicSwComponentType</a> , <a href="#">AtpBlueprint</a> , <a href="#">AtpBlueprintable</a> , <a href="#">AtpClassifier</a> , <a href="#">AtpType</a> , <a href="#">CollectableElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a> , <a href="#">SwComponentType</a>			
<b>Aggregated by</b>	<a href="#">ARPackage.element</a>			
Attribute	Type	Mult.	Kind	Note
—	—	—	—	—

**Table B.3: ApplicationSwComponentType**

Class	AssemblySwConnector			
<b>Note</b>	AssemblySwConnectors are exclusively used to connect SwComponentPrototypes in the context of a CompositionSwComponentType.			
<b>Base</b>	<a href="#">ARObject</a> , <a href="#">AtpClassifier</a> , <a href="#">AtpFeature</a> , <a href="#">AtpStructureElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">Referrable</a> , <a href="#">SwConnector</a>			
<b>Aggregated by</b>	<a href="#">AtpClassifier.atpFeature</a> , <a href="#">CompositionSwComponentType.connector</a>			
Attribute	Type	Mult.	Kind	Note
provider	AbstractProvidedPort Prototype	0..1	iref	Instance of providing port. <b>InstanceRef implemented by:</b> PPortInComposition InstanceRef
requester	AbstractRequiredPort Prototype	0..1	iref	Instance of requiring port. <b>InstanceRef implemented by:</b> RPortInComposition InstanceRef

**Table B.4: AssemblySwConnector**

<b>Class</b>	<b>AtomicSwComponentType</b> (abstract)			
<b>Note</b>	An atomic software component is atomic in the sense that it cannot be further decomposed and distributed across multiple ECUs.			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">AtpBlueprint</a> , <a href="#">AtpBlueprintable</a> , <a href="#">AtpClassifier</a> , <a href="#">AtpType</a> , <a href="#">CollectableElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a> , <a href="#">SwComponentType</a>			
<b>Subclasses</b>	<a href="#">ApplicationSwComponentType</a> , <a href="#">ComplexDeviceDriverSwComponentType</a> , <a href="#">EcuAbstractionSwComponentType</a> , <a href="#">NvBlockSwComponentType</a> , <a href="#">SensorActuatorSwComponentType</a> , <a href="#">ServiceProxySwComponentType</a> , <a href="#">ServiceSwComponentType</a>			
<b>Aggregated by</b>	<a href="#">ARPackage.element</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
internalBehavior	SwcInternalBehavior	0..1	aggr	The SwcInternalBehaviors owned by an AtomicSwComponentType can be located in a different physical file. Therefore the aggregation is <<atpSplitable>>. <b>Stereotypes:</b> atpSplitable; atpVariation <b>Tags:</b> atp.Splitkey=internalBehavior.shortName, internalBehavior.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
symbolProps	SymbolProps	0..1	aggr	This represents the SymbolProps for the AtomicSwComponentType. <b>Stereotypes:</b> atpSplitable <b>Tags:</b> atp.Splitkey=symbolProps.shortName

**Table B.5: AtomicSwComponentType**

<b>Class</b>	<b>CompositionSwComponentType</b>			
<b>Note</b>	A CompositionSwComponentType aggregates SwComponentPrototypes (that in turn are typed by <a href="#">SwComponentType</a> s) as well as SwConnectors for primarily connecting SwComponentPrototypes among each others and towards the surface of the CompositionSwComponentType. By this means, a hierarchical structures of software-components can be created. <b>Tags:</b> atp.recommendedPackage=SwComponentTypes			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">AtpBlueprint</a> , <a href="#">AtpBlueprintable</a> , <a href="#">AtpClassifier</a> , <a href="#">AtpType</a> , <a href="#">CollectableElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a> , <a href="#">SwComponentType</a>			
<b>Aggregated by</b>	<a href="#">ARPackage.element</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
component	SwComponentPrototype	*	aggr	The instantiated components that are part of this composition. The aggregation of SwComponentPrototype is subject to variability with the purpose to support the conditional existence of a SwComponentPrototype. Please be aware: if the conditional existence of SwComponentPrototypes is resolved post-build, the deselected SwComponentPrototypes are still contained in the ECUs build but the instances are inactive in that they are not scheduled by the RTE. The aggregation is marked as atpSplitable in order to allow the addition of service components to the ECU extract during the ECU integration. The use case for having 0 components owned by the CompositionSwComponentType could be to deliver an empty CompositionSwComponentType to e.g. a supplier for filling the internal structure. <b>Stereotypes:</b> atpSplitable; atpVariation <b>Tags:</b> atp.Splitkey=component.shortName, component.variationPoint.shortLabel vh.latestBindingTime=postBuild





Class	CompositionSwComponentType			
connector	SwConnector	*	aggr	<p>SwConnectors have the principal ability to establish a connection among <a href="#">PortPrototypes</a>. They can have many roles in the context of a <a href="#">CompositionSwComponentType</a>. Details are refined by subclasses.</p> <p>The aggregation of SwConnectors is subject to variability with the purpose to support variant data flow. The aggregation is marked as atpSplitable in order to allow the extension of the ECU extract with <a href="#">AssemblySwConnectors</a> between <a href="#">ApplicationSwComponentTypes</a> and <a href="#">ServiceSwComponentTypes</a> during the ECU integration.</p> <p><b>Stereotypes:</b> atpSplitable; atpVariation</p> <p><b>Tags:</b>  atp.Splitkey=connector.shortName, connector.variationPoint.shortLabel  vh.latestBindingTime=postBuild</p>
constantValue Mapping	ConstantSpecification MappingSet	*	ref	<p>Reference to the <a href="#">ConstantSpecificationMapping</a> to be applied for initValues of <a href="#">PPortComSpecs</a> and <a href="#">RPortComSpec</a>.</p> <p><b>Stereotypes:</b> atpSplitable</p> <p><b>Tags:</b> atp.Splitkey=constantValueMapping</p>
data Type Mapping	DataTypeMappingSet	*	ref	<p>Reference to the <a href="#">DataTypeMappingSet</a> to be applied for the used <a href="#">ApplicationDataTypes</a> in <a href="#">PortInterfaces</a>.</p> <p>Background: when developing subsystems it may happen that <a href="#">ApplicationDataTypes</a> are used on the surface of <a href="#">CompositionSwComponentTypes</a>. In this case it would be reasonable to be able to also provide the intended mapping to the <a href="#">ImplementationDataTypes</a>. However, this mapping shall be informal and not technically binding for the implementors mainly because the RTE generator is not concerned about the <a href="#">CompositionSwComponentTypes</a>.</p> <p>Rationale: if the mapping of <a href="#">ApplicationDataTypes</a> on the delegated and inner <a href="#">PortPrototype</a> matches then the mapping to <a href="#">ImplementationDataTypes</a> is not impacting compatibility.</p> <p><b>Stereotypes:</b> atpSplitable</p> <p><b>Tags:</b> atp.Splitkey=dataTypeMapping</p>
instantiation RTEEventProps	InstantiationRTEEvent Props	*	aggr	<p>This allows to define instantiation specific properties for RTE Events, in particular for instance specific scheduling.</p> <p><b>Stereotypes:</b> atpSplitable; atpVariation</p> <p><b>Tags:</b>  atp.Splitkey=instantiationRTEEventProps.shortLabel, instantiationRTEEventProps.variationPoint.shortLabel  vh.latestBindingTime=codeGenerationTime</p> <p>This Attribute is only used by the AUTOSAR Classic Platform.</p>
physical Dimension Mapping	PhysicalDimension MappingSet	0..1	ref	<p>This reference identifies the <a href="#">PhysicalDimensionMappingSet</a> that is applicable in the context of the enclosing <a href="#">CompositionSwComponentType</a>. The <a href="#">PhysicalDimensionMappings</a> contained in the <a href="#">PhysicalDimensionMappingSet</a> shall be taken into account for the assessment of the compatibility of <a href="#">PhysicalDimensions</a> in the context of creation of a <a href="#">PortInterfaceMapping</a> in the scope of the <a href="#">CompositionSwComponentType</a>.</p>

**Table B.6: CompositionSwComponentType**

<b>Class</b>	<b>DataMapping</b> (abstract)			
<b>Note</b>	Mapping of port elements (data elements and parameters) to frames and signals.			
<b>Base</b>	<i>ARObject</i>			
<b>Subclasses</b>	ClientServerToSignalMapping, SenderReceiverCompositeElementToSignalMapping, SenderReceiverToSignalGroupMapping, <a href="#">SenderReceiverToSignalMapping</a> , TriggerToSignalMapping			
<b>Aggregated by</b>	SystemMapping.dataMapping			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
introduction	DocumentationBlock	0..1	aggr	This represents introductory documentation about the data mapping.

**Table B.7: DataMapping**

<b>Class</b>	<b>EcuAbstractionSwComponentType</b>			
<b>Note</b>	The ECUAbstraction is a special AtomicSwComponentType that resides between a software-component that wants to access ECU periphery and the Microcontroller Abstraction. The EcuAbstractionSw ComponentType introduces the possibility to link from the software representation to its hardware description provided by the ECU Resource Template. <b>Tags:</b> atp.recommendedPackage=SwComponentTypes			
<b>Base</b>	<a href="#">ARElement</a> , <i>ARObject</i> , <a href="#">AtomicSwComponentType</a> , <i>AtpBlueprint</i> , <i>AtpBlueprintable</i> , <i>AtpClassifier</i> , <i>AtpType</i> , <i>CollectableElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>PackageableElement</i> , <i>Referrable</i> , <a href="#">SwComponentType</a>			
<b>Aggregated by</b>	<a href="#">ARPackage.element</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
hardware Element	HwDescriptionEntity	*	ref	Reference from the EcuAbstractionComponentType to the description of the used HwElements.

**Table B.8: EcuAbstractionSwComponentType**

<b>Class</b>	<b>InternalBehavior</b> (abstract)			
<b>Note</b>	Common base class (abstract) for the internal behavior of both software components and basic software modules/clusters.			
<b>Base</b>	<i>ARObject</i> , <i>AtpClassifier</i> , <i>AtpFeature</i> , <i>AtpStructureElement</i> , <i>Identifiable</i> , <i>MultilanguageReferrable</i> , <i>Referrable</i>			
<b>Subclasses</b>	BswInternalBehavior, SwcInternalBehavior			
<b>Aggregated by</b>	<i>AtpClassifier.atpFeature</i>			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
constant Memory	ParameterData Prototype	*	aggr	Describes a read only memory object containing characteristic value(s) implemented by this Internal Behavior. The shortName of ParameterDataPrototype has to be equal to the 'C' identifier of the described constant. The characteristic value(s) might be shared between Sw ComponentPrototypes of the same SwComponentType. The aggregation of constantMemory is subject to variability with the purpose to support variability in the software component or module implementations. Typically different algorithms in the implementation are requiring different number of memory objects. <b>Stereotypes:</b> atp.Splitable; atp.Variation <b>Tags:</b> atp.Splitkey=constantMemory.shortName, constantMemory.variationPoint.shortLabel vh.latestBindingTime=preCompileTime





Class	InternalBehavior (abstract)			
constantValueMapping	ConstantSpecificationMappingSet	*	ref	Reference to the ConstantSpecificationMapping to be applied for the particular InternalBehavior <b>Stereotypes:</b> atpSplitable <b>Tags:</b> atp.Splitkey=constantValueMapping
dataTypeMapping	DataTypeMappingSet	*	ref	Reference to the DataTypeMapping to be applied for the particular InternalBehavior <b>Stereotypes:</b> atpSplitable <b>Tags:</b> atp.Splitkey=dataTypeMapping
exclusiveArea	ExclusiveArea	*	aggr	This specifies an ExclusiveArea for this InternalBehavior. The exclusiveArea is local to the component resp. module. The aggregation of ExclusiveAreas is subject to variability. Note: the number of ExclusiveAreas might vary due to the conditional existence of RunnableEntities or BswModuleEntities. <b>Stereotypes:</b> atpSplitable; atpVariation <b>Tags:</b> atp.Splitkey=exclusiveArea.shortName, exclusiveArea.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
exclusiveAreaNestingOrder	ExclusiveAreaNestingOrder	*	aggr	This represents the set of ExclusiveAreaNestingOrder owned by the InternalBehavior. <b>Stereotypes:</b> atpSplitable; atpVariation <b>Tags:</b> atp.Splitkey=exclusiveAreaNestingOrder.shortName, exclusiveAreaNestingOrder.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
staticMemory	<a href="#">VariableDataPrototype</a>	*	aggr	Describes a read and writeable static memory object representing measurement variables implemented by this software component. The term "static" is used in the meaning of "non-temporary" and does not necessarily specify a linker encapsulation. This kind of memory is only supported if supportsMultipleInstantiation is FALSE. The shortName of the VariableDataPrototype has to be equal with the "C" identifier of the described variable. The aggregation of staticMemory is subject to variability with the purpose to support variability in the software component's implementations. Typically different algorithms in the implementation are requiring different number of memory objects. <b>Stereotypes:</b> atpSplitable; atpVariation <b>Tags:</b> atp.Splitkey=staticMemory.shortName, staticMemory.variationPoint.shortLabel vh.latestBindingTime=preCompileTime

**Table B.9: InternalBehavior**

Class	PortInterface (abstract)			
Note	Abstract base class for an interface that is either provided or required by a port of a software component.			
Base	<a href="#">ARElement</a> , ARObject, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Subclasses	ClientServerInterface, DataInterface, ModeSwitchInterface, TriggerInterface			
Aggregated by	<a href="#">ARPackage.element</a>			
Attribute	Type	Mult.	Kind	Note





Class	PortInterface (abstract)			
isService	Boolean	0..1	attr	<p>This flag is set if the PortInterface is to be used for communication between an</p> <ul style="list-style-type: none"> <li>• ApplicationSwComponentType or</li> <li>• ServiceProxySwComponentType or</li> <li>• SensorActuatorSwComponentType or</li> <li>• ComplexDeviceDriverSwComponentType</li> <li>• ServiceSwComponentType</li> <li>• EcuAbstractionSwComponentType</li> </ul> <p>and a ServiceSwComponentType (namely an AUTOSAR Service) located on the same ECU. Otherwise the flag is not set.</p> <p><b>Stereotypes:</b> atpVariation  <b>Tags:</b> vh.latestBindingTime=blueprintDerivationTime</p> <p>This Attribute is only used by the AUTOSAR Classic Platform.</p>
serviceKind	ServiceProviderEnum	0..1	attr	<p>This attribute provides further details about the nature of the applied service.</p> <p>This Attribute is only used by the AUTOSAR Classic Platform.</p>

Table B.10: PortInterface

Class	PortPrototype (abstract)			
<b>Note</b>	<p>Base class for the ports of an AUTOSAR software component.</p> <p>The aggregation of PortPrototypes is subject to variability with the purpose to support the conditional existence of ports.</p>			
<b>Base</b>	ARObject, AtpBlueprintable, AtpFeature, AtpPrototype, Identifiable, MultilanguageReferrable, Referrable			
<b>Subclasses</b>	AbstractProvidedPortPrototype, AbstractRequiredPortPrototype			
<b>Aggregated by</b>	AtpClassifier.atpFeature, <a href="#">SwComponentType.port</a>			
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.
senderReceiverAnnotation	SenderReceiverAnnotation	*	aggr	Collection of annotations of this ports sender/receiver communication.
triggerPortAnnotation	TriggerPortAnnotation	*	aggr	Annotations on this trigger port.

Table B.11: PortPrototype



<b>Class</b>	<b>RootSwCompositionPrototype</b>			
<b>Note</b>	<p>The RootSwCompositionPrototype represents the top-level-composition of software components within a given System.</p> <p>According to the use case of the System, this may for example be a more or less complete VFB description, the software of a System Extract or the software of a flat ECU Extract with only atomic SWCs. Therefore the RootSwComposition will only occasionally contain all atomic software components that are used in a complete VFB System. The OEM is primarily interested in the required functionality and the interfaces defining the integration of the Software Component into the System. The internal structure of such a component contains often substantial intellectual property of a supplier. Therefore a top-level software composition will often contain empty compositions which represent subsystems.</p> <p>The contained SwComponentPrototypes are fully specified by their SwComponentTypes (including Port Prototypes, PortInterfaces, VariableDataPrototypes, SwcInternalBehavior etc.), and their ports are interconnected using SwConnectorPrototypes.</p>			
<b>Base</b>	ARObject, AtpFeature, AtpPrototype, Identifiable, MultilanguageReferrable, Referrable			
<b>Aggregated by</b>	AtpClassifier.atpFeature, <a href="#">System.rootSoftwareComposition</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
calibrationParameterValueSet	CalibrationParameterValueSet	*	ref	<p>Used CalibrationParameterValueSet for instance specific initialization of calibration parameters.</p> <p><b>Stereotypes:</b> atpSplitable</p> <p><b>Tags:</b> atp.Splitkey=calibrationParameterValueSet</p> <p>This Attribute is only used by the AUTOSAR Classic Platform.</p>
flatMap	FlatMap	0..1	ref	<p>The FlatMap used in the scope of this RootSwCompositionPrototype.</p> <p><b>Stereotypes:</b> atpSplitable</p> <p><b>Tags:</b> atp.Splitkey=flatMap</p> <p>This Attribute is only used by the AUTOSAR Classic Platform.</p>
softwareComposition	<a href="#">CompositionSwComponentType</a>	0..1	trf	<p>We assume that there is exactly one top-level composition that includes all Component instances of the system.</p> <p><b>Stereotypes:</b> isOfType</p>

**Table B.12: RootSwCompositionPrototype**

<b>Class</b>	<b>SenderReceiverToSignalMapping</b>			
<b>Note</b>	Mapping of a sender receiver communication data element to a signal.			
<b>Base</b>	ARObject, <a href="#">DataMapping</a>			
<b>Aggregated by</b>	SystemMapping.dataMapping			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
dataElement	<a href="#">VariableDataPrototype</a>	0..1	iref	<p>Reference to the data element.</p> <p><b>InstanceRef implemented by:</b> VariableDataPrototypeInSystemInstanceRef</p>
senderToSignalTextTableMapping	TextTableMapping	0..1	aggr	This mapping allows for the text-table translation between the sending DataPrototype that is defined in the Port Prototype and the physicalProps defined for the System Signal.
signalToReceiverTextTableMapping	TextTableMapping	0..1	aggr	This mapping allows for the text-table translation between the physicalProps defined for the SystemSignal and a receiving DataPrototype that is defined in the Port Prototype.
systemSignal	SystemSignal	0..1	ref	Reference to the system signal used to carry the data element.

**Table B.13: SenderReceiverToSignalMapping**



<b>Class</b>	<b>SensorActuatorSwComponentType</b>			
<b>Note</b>	The SensorActuatorSwComponentType introduces the possibility to link from the software representation of a sensor/actuator to its hardware description provided by the ECU Resource Template. <b>Tags:</b> atp.recommendedPackage=SwComponentTypes			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">AtomicSwComponentType</a> , <a href="#">AtpBlueprint</a> , <a href="#">AtpBlueprintable</a> , <a href="#">AtpClassifier</a> , <a href="#">AtpType</a> , <a href="#">CollectableElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a> , <a href="#">SwComponentType</a>			
<b>Aggregated by</b>	<a href="#">ARPackage.element</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
sensorActuator	HwDescriptionEntity	0..1	ref	Reference from the Sensor Actuator Software Component Type to the description of the actual hardware.

**Table B.14: SensorActuatorSwComponentType**

<b>Class</b>	<b>SwComponentType</b> (abstract)			
<b>Note</b>	Base class for AUTOSAR software components.			
<b>Base</b>	<a href="#">ARElement</a> , <a href="#">ARObject</a> , <a href="#">AtpBlueprint</a> , <a href="#">AtpBlueprintable</a> , <a href="#">AtpClassifier</a> , <a href="#">AtpType</a> , <a href="#">CollectableElement</a> , <a href="#">Identifiable</a> , <a href="#">MultilanguageReferrable</a> , <a href="#">PackageableElement</a> , <a href="#">Referrable</a>			
<b>Subclasses</b>	<a href="#">AtomicSwComponentType</a> , <a href="#">CompositionSwComponentType</a> , <a href="#">ParameterSwComponentType</a>			
<b>Aggregated by</b>	<a href="#">ARPackage.element</a>			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
consistency Needs	ConsistencyNeeds	*	aggr	This represents the collection of ConsistencyNeeds owned by the enclosing SwComponentType. <b>Stereotypes:</b> atpSplitable; atpVariation <b>Tags:</b> atp.Splitkey=consistencyNeeds.shortName, consistencyNeeds.variationPoint.shortLabel vh.latestBindingTime=preCompileTime This Attribute is only used by the AUTOSAR Classic Platform.
port	<a href="#">PortPrototype</a>	*	aggr	The PortPrototypes through which this SwComponent Type can communicate. The aggregation of PortPrototype is subject to variability with the purpose to support the conditional existence of PortPrototypes. <b>Stereotypes:</b> atpSplitable; atpVariation <b>Tags:</b> atp.Splitkey=port.shortName, port.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
portGroup	PortGroup	*	aggr	A port group being part of this component. <b>Stereotypes:</b> atpSplitable; atpVariation <b>Tags:</b> atp.Splitkey=portGroup.shortName, portGroup.variationPoint.shortLabel vh.latestBindingTime=preCompileTime
swcMapping Constraint	SwComponentMapping Constraints	*	ref	Reference to constraints that are valid for this Sw ComponentType. This Attribute is only used by the AUTOSAR Classic Platform.
swComponent Documentation	SwComponent Documentation	0..1	aggr	This adds a documentation to the SwComponentType. <b>Stereotypes:</b> atpSplitable; atpVariation <b>Tags:</b> atp.Splitkey=swComponentDocumentation, swComponentDocumentation.variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=-10





Class	SwComponentType (abstract)			
unitGroup	UnitGroup	*	ref	This allows for the specification of which UnitGroups are relevant in the context of referencing SwComponentType. This Attribute is only used by the AUTOSAR Classic Platform.

**Table B.15: SwComponentType**

Class	System			
<b>Note</b>	The top level element of the System Description. The System description defines five major elements: Topology, Software, Communication, Mapping and Mapping Constraints. The System element directly aggregates the elements describing the Software, Mapping and Mapping Constraints; it contains a reference to an ASAM FIBEX description specifying Communication and Topology. <b>Tags:</b> atp.recommendedPackage=Systems			
<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="#">UploadableDesignElement</a> , <a href="#">UploadablePackageElement</a>			
<b>Aggregated by</b>	<a href="#">ARPackage.element</a> , <a href="#">AtpClassifier.atpFeature</a>			
Attribute	Type	Mult.	Kind	Note
clientId DefinitionSet	ClientIdDefinitionSet	*	ref	Set of Client Identifiers that are used for inter-ECU client-server communication in the System. This Attribute is only used by the AUTOSAR Classic Platform.
containerIPdu HeaderByte Order	ByteOrderEnum	0..1	attr	Defines the byteOrder of the header in ContainerIPdus. This Attribute is only used by the AUTOSAR Classic Platform.
ecuExtract Version	RevisionLabelString	0..1	attr	Version number of the Ecu Extract. This Attribute is only used by the AUTOSAR Classic Platform.
fibexElement	FibexElement	*	ref	Reference to ASAM FIBEX elements specifying Communication and Topology. All Fibex Elements used within a System Description shall be referenced from the System Element. atpVariation: In order to describe a product-line, all Fibex Elements can be optional. <b>Stereotypes:</b> atpSplitable; atpVariation <b>Tags:</b> atp.Splitkey=fibexElement.fibexElement, fibexElement.variationPoint.shortLabel vh.latestBindingTime=postBuild
interpolation Routine MappingSet	InterpolationRoutineMappingSet	*	ref	This reference identifies the InterpolationRoutineMapping Sets that are relevant in the context of the enclosing System. This Attribute is only used by the AUTOSAR Classic Platform.
j1939Shared AddressCluster	J1939SharedAddressCluster	*	aggr	Collection of J1939Clusters that share a common address space for the routing of messages. <b>Stereotypes:</b> atpSplitable; atpVariation <b>Tags:</b> atp.Splitkey=j1939SharedAddressCluster.shortName, j1939SharedAddressCluster.variationPoint.shortLabel vh.latestBindingTime=postBuild This Attribute is only used by the AUTOSAR Classic Platform.





Class	System			
mapping	SystemMapping	*	aggr	<p>Aggregation of all mapping aspects (mapping of SW components to ECUs, mapping of data elements to signals, and mapping constraints). In order to support OEM / Tier 1 interaction and shared development for one common System this aggregation is atpSplitable and atpVariation. The content of System Mapping can be provided by several parties using different names for the SystemMapping. This element is not required when the System description is used for a network-only use-case.</p> <p><b>Stereotypes:</b> atpSplitable; atpVariation</p> <p><b>Tags:</b>  atp.Splitkey=mapping.shortName, mapping.variationPoint.shortLabel  vh.latestBindingTime=postBuild</p>
pncVectorLength	PositiveInteger	0..1	attr	Length of the partial networking request release information vector (in bytes).
pncVectorOffset	PositiveInteger	0..1	attr	Absolute offset (with respect to the NM-PDU) of the partial networking request release information vector that is defined in bytes as an index starting with 0.
rootSoftwareComposition	<a href="#">RootSwCompositionPrototype</a>	0..1	aggr	<p>Aggregation of the root software composition, containing all software components in the System in a hierarchical structure. This element is not required when the System description is used for a network-only use-case.</p> <p>atpVariation: The RootSwCompositionPrototype can vary.</p> <p><b>Stereotypes:</b> atpSplitable; atpVariation</p> <p><b>Tags:</b>  atp.Splitkey=rootSoftwareComposition.shortName, rootSoftwareComposition.variationPoint.shortLabel  vh.latestBindingTime=systemDesignTime  This Attribute is only used by the AUTOSAR Classic Platform.</p>
swCluster	CpSoftwareCluster	*	ref	<p>CP Software Clusters of this System</p> <p><b>Stereotypes:</b> atpSplitable; atpVariation</p> <p><b>Tags:</b>  atp.Splitkey=swCluster.cpSoftwareCluster, swCluster.variationPoint.shortLabel  vh.latestBindingTime=systemDesignTime  This Attribute is only used by the AUTOSAR Classic Platform.</p>
systemDocumentation	Chapter	*	aggr	<p>Possibility to provide additional documentation while defining the System. The System documentation can be composed of several chapters.</p> <p><b>Stereotypes:</b> atpSplitable; atpVariation</p> <p><b>Tags:</b>  atp.Splitkey=systemDocumentation.shortName, systemDocumentation.variationPoint.shortLabel  vh.latestBindingTime=systemDesignTime  xml.sequenceOffset=-10  This Attribute is only used by the AUTOSAR Classic Platform.</p>
systemVersion	RevisionLabelString	0..1	attr	Version number of the System Description.

**Table B.16: System**

Class	VariableDataPrototype
<b>Note</b>	A VariableDataPrototype represents a formalized generic piece of information that is typically mutable by the application software layer. VariableDataPrototype is used in various contexts and the specific context gives the otherwise generic VariableDataPrototype a dedicated semantics.





<b>Class</b>	<b>VariableDataPrototype</b>			
<b>Base</b>	<i>ARObject, AtpFeature, AtpPrototype, AutosarDataPrototype, DataPrototype, Identifiable, Multilanguage Referrable, Referrable</i>			
<b>Aggregated by</b>	ApplicationInterface.indication, <i>AtpClassifier.atpFeature</i> , BswInternalBehavior.arTypedPerInstanceMemory, BswModuleDescription.providedData, BswModuleDescription.requiredData, BulkNvDataDescriptor.bulkNvBlock, <i>InternalBehavior.staticMemory</i> , NvBlockDescriptor.ramBlock, NvDataInterface.nvData, SenderReceiverInterface.dataElement, ServiceInterface.event, SwcInternalBehavior.arTypedPerInstanceMemory, SwcInternalBehavior.explicitInterRunnableVariable, SwcInternalBehavior.implicitInterRunnableVariable			
<b>Attribute</b>	<b>Type</b>	<b>Mult.</b>	<b>Kind</b>	<b>Note</b>
initValue	ValueSpecification	0..1	aggr	Specifies initial value(s) of the VariableDataPrototype

**Table B.17: VariableDataPrototype**